THE APPLICATION OF EXTENDED KALMAN FILTERING TO THE POSITION LOCATING REPORTING SYSTEM (PLRS)

Charles A. Dittmar

Report Number — NPS-52TS75121

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE APPLICATION OF EXTENDED
KALMAN FILTERING TO THE
POSITION LOCATING REPORTING SYSTEM (PLRS)

by

Charles A. Dittmar, Jr.

December 1975

Thesis Advisor: H.A. Titus

Approved for public release; distribution unlimited.

Prepared for:
Naval Electronics System Command
Washington, D.C. 22030

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Isham Linder
Superintendent

Jack R. Borsting
Provost

This thesis prepared in conjunction with research supported in part by Naval Electronics System Command under Project Number N0003975PO52945.

Released as a
Technical Report by:

Dean of Research

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS-52TS75121	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Application of Extended Kalman Filtering to the Position Locating Reporting System (PLRS)		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Charles A. Dittmar, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1975 13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Research supported in part by Naval Electronics System Command under Project Number N0003975PO52945.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

PLRS
Position Locating Reporting System
Extended Kalman filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Extended Kalman filtering is applied to the PLRS (Position Locating Reporting System). Here the nonlinearity to the filter enters through the measurement (range only). The nonlinearity being the relationship between range and the cartesian coordinate states of the filter.

Filter covariances of error are portrayed as error ellipsoids. These are used to determine which one

(20. ABSTRACT Continued)

should be used to update a unit when there are several other units available for ranging. One should attempt to make the range measurement in the same direction as the major axis of error associated with the unit to be updated.

The filtering techniques are evaluated using static units and high speed maneuvering aircraft.

The Application of Extended Kalman Filtering to the

Position Locating Reporting System(PLRS)

by

Charles A. Dittmar, Jr.
Captain, United States Marine Corps
B.S., United States Naval Academy, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1975

ABSTRACT

Extended Kalman filtering is applied to the PLRS (Position Locating Reporting System). Here the nonlinearity to the filter enters through the measurement (range only). The nonlinearity being the relationship between range and the cartesian coordinate states of the filter.

Filter covariances of error are portrayed as error ellipsoids. These are used to determine which one should be used to update a unit when there are several other units available for ranging. One should attempt to make the range measurement in the same direction of the major axis of error associated with the unit to be updated.

The filtering techniques are evaluated using static units and high speed maneuvering aircraft.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Position location of tactical elements in the Marine Corps in the past has relied upon the fact that these units had maps from which they could determine their position. Position determination is therefore highly dependent on the ability of an individual to correctly orient himself to the natural terrain surrounding him and locate this same terrain on the map. Once located, the requirement then exists to transmit position information via radio link to parent units. Meshed with the ground commander's problem of position location is the task of organizing and conducting support operations such as medical evacuation of wounded personnel, artillery, naval gunfire and air support.

Using visual references it becomes difficult to accurately locate one's position when operating in mountainous terrain, in heavy vegetation, at night or in bad weather. And with the heavy volume of radio traffic experienced during battle, the probability of communicating to higher headquarters a unit's position or information about that position is reduced. As a result the support operations mentioned earlier are delayed because of detailed clearances required to ensure protection of friendly units whose position is not precisely known.

The obstacles to position location suggested can and do arise in the conduct of actual operations, and for that

reason, the Marine Corps has written specifications detailing a Position Location and Reporting System (PLRS) designed for tactical employment on the battlefield, [1]. The system as construed is to consist of a containerized master unit composed of a general purpose computer and communications control electronics. This unit is referred to as the Master Unit. It is responsible for processing all range information inputs to the system, and displaying unit positions on a visual display with a variety of other information.

The other units in the area are all portable, capable of being carried by a single man, vehicle, or aircraft. They are referred to as update or ranging units depending on their particular function at the time. When a unit's position is being improved, that unit is referred to as the "update" unit. All other units that are in contact with the update unit and are providing range measurements are "ranging" units. The frequency at which each unit is updated is specified according to its type. That is, aircraft traveling at much higher velocities than men will require updates more often if their tracking is to be accurate.

Before a unit can be tracked and position estimates made, an initial estimate of position must be made. This process of initialization is controlled by the master unit. Once a unit is initialized the computer systematically selects ranging units processing their associated range information as applicable. In processing range information a new position estimate and a measure of the uncertainty in the estimate is provided.

While the processing of accumulated range information for an update unit appears to be straightforward, the difficulty lies in deciding which ranges to process so that the most accurate position estimates will be obtained. Range information is a measurement of the time required to send a signal from the ranging unit to the update unit and back again plus some built in delays.

While the measurement of times associated with the total time is quite accurate, it is still subject to the effects of noise therefore range is a noisy measurement. In addition the uncertainty of the ranging unit's position adds to the uncertainty of the measurement itself. Therefore the selection of a ranging unit to best update a particular unit becomes a very important factor in the position location problem. One would desire to use the ranging unit nearest the line of the major axis of error in the unit to be updated.

The error ellipsoids of the "update" and "ranging" units and their relative geometry are analyzed and their influence on the tracking is portrayed in examples with maneuvering high speed aircraft and with stationary units.

II. POSITION LOCATING PROBLEM

A. THE FILTER'S DYNAMIC MODEL OF AIRCRAFT UNITS

First consider units in aircraft in a two dimensional space.* The estimate that we seek in this problem is position (X,Y) and velocity, $(\dot{X},\dot{Y})$. X and Y are calculated with respect to a relative grid at a point on the earth's surface. The altitude is the vertical height above the earth at that particular XY grid intersection, and is not utilized in this investigation.

For tracking we may choose a fourth order state vector,

$$\underline{x} = \begin{bmatrix} X \\ \dot{X} \\ Y \\ \dot{Y} \end{bmatrix} \tag{1}$$

Writing the equations in linear state form results in

$$\underline{x}(k+1) = \Phi\, \underline{x}(k) + \Gamma\, \underline{w}(k) \tag{2}$$

where

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

---

*The altitude tracking will be handled independently.

and

$$\Gamma = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \tag{4}$$

B. EXTENDED KALMAN FILTER

The observable range is a nonlinear function of the states. So we must briefly look at how the nonlinearities can be handled in this type of problem.

Consider a nonlinear discrete system of state and observation equations given by

$$\underline{x}(k+1) = \underline{f}(\underline{x}(k),k) + \underline{g}(\underline{x}(k),k)\cdot\underline{w}(k) \tag{5}$$

and

$$\underline{z}(k) = \underline{h}(\underline{x}(k),k) + \underline{v}(k) \quad . \tag{6}$$

In these equations $\underline{f}$, $\underline{g}$ and $\underline{h}$ are nonlinear functions of the state variables $\underline{x}$, $\underline{w}(k)$ is plant excitation noise, and $\underline{v}(k)$ is measurement noise. The plant noise and measurement noise are assumed uncorrelated, zero-mean, and white. That is

$$E[\underline{w}(k) \cdot \underline{w}^T(j)] = Q'(k)\, \delta_{kj}$$

and

$$E[\underline{v}(k) \cdot \underline{v}^T(j)] = R(k)\, \delta_{kj}$$

In order to apply the linear filter equations, (5) and (6) are expanded about the best estimate of the state at that time and only the first-order terms are kept. Equation (5) gives

$$\underline{x}(k+1) = \Phi(k)\, \underline{x}(k) + \Gamma(k) \cdot \underline{w}(k) \quad , \qquad (7)$$

with

$$\Phi(k) = \left. \frac{\partial \underline{f}}{\partial \underline{x}} \right|_{\underline{x}=\hat{\underline{x}}(k)} .$$

Similarly Eq. (6) yields

$$\underline{z}(k) = H(k)\, \underline{x}(k) + \underline{v}(k) \qquad (8)$$

where

$$H(k) = \left. \frac{\partial \underline{h}}{\partial \underline{x}} \right|_{\underline{x}=\underline{x}'(k)} . \qquad (9)$$

$\hat{\underline{x}}(k)$ is the estimated state value after the k^{th} measurement and $\underline{x}'(k)$ is the predicted value of the state before the k^{th} measurement. That is,

$$\underline{x}'(k) = \underline{f}(\hat{\underline{x}}(k-1), k-1) \quad . \tag{10}$$

A state error vector is defined by

$$\tilde{\underline{x}}(k) = \hat{\underline{x}}(k) - \underline{x}(k) \quad ,$$

and a predicted state error vector is defined by

$$\tilde{\underline{x}}'(k) = \underline{x}'(k) - \underline{x}(k) \quad .$$

The covariance of state error matrix is defined by

$$P(k) = E[\tilde{\underline{x}}(k) \cdot \tilde{\underline{x}}^T(k)] \quad ,$$

and the predicted covariance of state error matrix is given by

$$P'(k) = E[\tilde{\underline{x}}'(k) \cdot \tilde{\underline{x}}'^T(k)] \quad .$$

The state excitation matrix is given by

$$Q(k) = E[\Gamma(k) \cdot \underline{w}(k) \cdot \underline{w}^T(k) \cdot \Gamma^T(k)] \quad ,$$

and the measurement noise covariance matrix is

$$R(k) = E[\underline{v}(k) \cdot \underline{v}^T(k)] \quad .$$

The Kalman Filter equations are given by [2]

$$P'(k+1) = \Phi(k)P(k)\Phi^T(k) + Q(k) \tag{11}$$

$$G(k) = P'(k)H^T(k)[H(k)P'(k)H^T(k) + R(k)]^{-1} \tag{12}$$

$$P(k) = [I - G(k)H(k)]\, P'(k) \tag{13}$$

$$\underline{x}'(k) = \underline{f}(\hat{\underline{x}}(k-1),k) \tag{14}$$

$$\underline{z}'(k) = \underline{h}(\underline{x}'(k),k) \tag{15}$$

$$\hat{\underline{x}}(k) = \underline{x}'(k) + G(k)[\underline{z}(k) - \underline{z}'(k)] \quad . \tag{16}$$

The Q matrix serves not only to allow for maneuvering but also to account for any model inaccuracies. That is, any discrepancies between the true action of the physical system and its characterization by Eq. (7). For a filter which reaches steady-state conditions the Q also serves to prevent the gain matrix G(k) from approaching zero by always insuring uncertainty in the predicted covariance of error matrix P'(k).

The observation equation is nonlinear in the states and is given by

$$z(k) = [x(k)^2 + y(k)^2]^{1/2} \tag{17}$$

Equation (9) can be used to give observation matrix. The result is

$$H(k) = \left[\frac{x(k)}{x(k)^2 + y(k)^2} \quad 0 \quad \frac{y(k)}{x(k)^2 + y(k)^2} \quad 0 \right] \tag{18}$$

In doing this, the recursive Kalman filter equations have been extended to cover the nonlinear case. The pseudo measurement matrix, H, must now be calculated from the position estimates at each time interval.

The use of the Extended Kalman technique in tracking a unit as it moves through the area presents a unique problem. Because we are observing only the range, we can improve our estimate of location only along the direction of the range vector. That is, the uncertainty or error variance in location remains unchanged in cross-range. We must then choose another ranging unit nearby that will improve upon this cross range error. This however is done later at the next prescribed interval for update of the unit in question.

The conclusion drawn from this observation is that it is critical that an algorithm be developed for selecting the best observations (ranging units) to process in the updating of a unit. It will be shown that this algorithm must be dependent on the physical geometry associated with the covariance error ellipses. It takes into consideration the error covariances associated with both the update and ranging units.

III. RANGE MEASUREMENT PROCESSING AND ERROR ELLIPSOIDS

The position of each unit within the system is established with a degree of uncertainty about the estimate. This uncertainty is expressed in the covariance of error matrix, P, and represents the sigma squared error deviation about the estimate. The position diagonal terms (P_{11} and P_{33}) of the covariance matrix represent the variances of the estimate in the XY coordinate directions. Their respective off diagonal terms (covariances) represent the degree of coupling and the orientation of the uncertainty in the XY plane.

If the errors were normally distributed, there exists a rotated coordinate system such that in the new system the orthogonal position components are uncorrelated. This is equivalent to taking the exponent of the joint normal probability density function, and applying a coordinate transformation which eliminates the cross terms.

The exponent (for zero-mean random variables) is

$$\frac{x^2}{\sigma_x^2} - \frac{2r_{xy}xy}{\sigma_x\sigma_y} + \frac{y^2}{\sigma_y^2} \quad . \tag{19}$$

When set equal to a constant, this curve, which is an ellipse, is a curve of constant probability. This ellipse does not have its major and minor axes aligned with the coordinate system however. By applying the transformation

$$x' = x \cos\theta + y \sin\theta \tag{20}$$

and

$$y' = y \cos\theta - x \sin\theta \tag{21}$$

with

$$\theta = \frac{1}{2} \tan^{-1} \left[\frac{2 \operatorname{cov}(x,y)}{\sigma_x^2 - \sigma_y^2} \right] , \tag{22}$$

the ellipse will be aligned with the x', y' axis and the resultant random variables will be uncorrelated. The new variances in this system are calculated by

$$\sigma_{x'}^2 = \frac{\sigma_x^2 + \sigma_y^2}{2} + \frac{\operatorname{cov}(xy)}{\sin 2\theta} , \tag{23}$$

and

$$\sigma_{y'}^2 = \frac{\sigma_x^2 + \sigma_y^2}{2} - \frac{\operatorname{cov}(xy)}{\sin 2\theta} . \tag{24}$$

This method of viewing the sample statistics provides insight into the filter performance because it shows the regions of high probability and their geometric relationships. The values of $\sigma_{x'}^2$ and $\sigma_{y'}^2$ are referred to as the rotated error variances in the following section.

When a range measurement is made between two units, each with a different and independent error ellipse, the uncertainty of position along the range line connecting the two is improved for both units. In addition, it is sometimes possible to reduce the error of the axis orthogonal to the range line depending on the orientation of the error ellipse.

To study the influence of the error ellipsoids of the ranging and update unit let us consider a static example. This model is synomymous with those used for foot soldiers.

Measurement noise will be considered to come solely from the error variance of the ranging unit and vice versa when their roles interchanged. That is, at the same time the "update" unit is being updated, let us also update the "ranging" unit. The estimates of position and their error ellipsoids are shown in Fig. 1.

A projection of the error ellipsoid along the y-axis is given in Fig. 2. If one utilizes each as an observation to update the other, the resulting variance of both along the range line is given by

$$\sigma_{new}^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

where σ_1 and σ_2 are the standard deviations of units 1 and 2 in the range projection and σ_{new} is the new updated standard deviation for both units in the range projection (σ_y), as is seen also in Fig. 2.

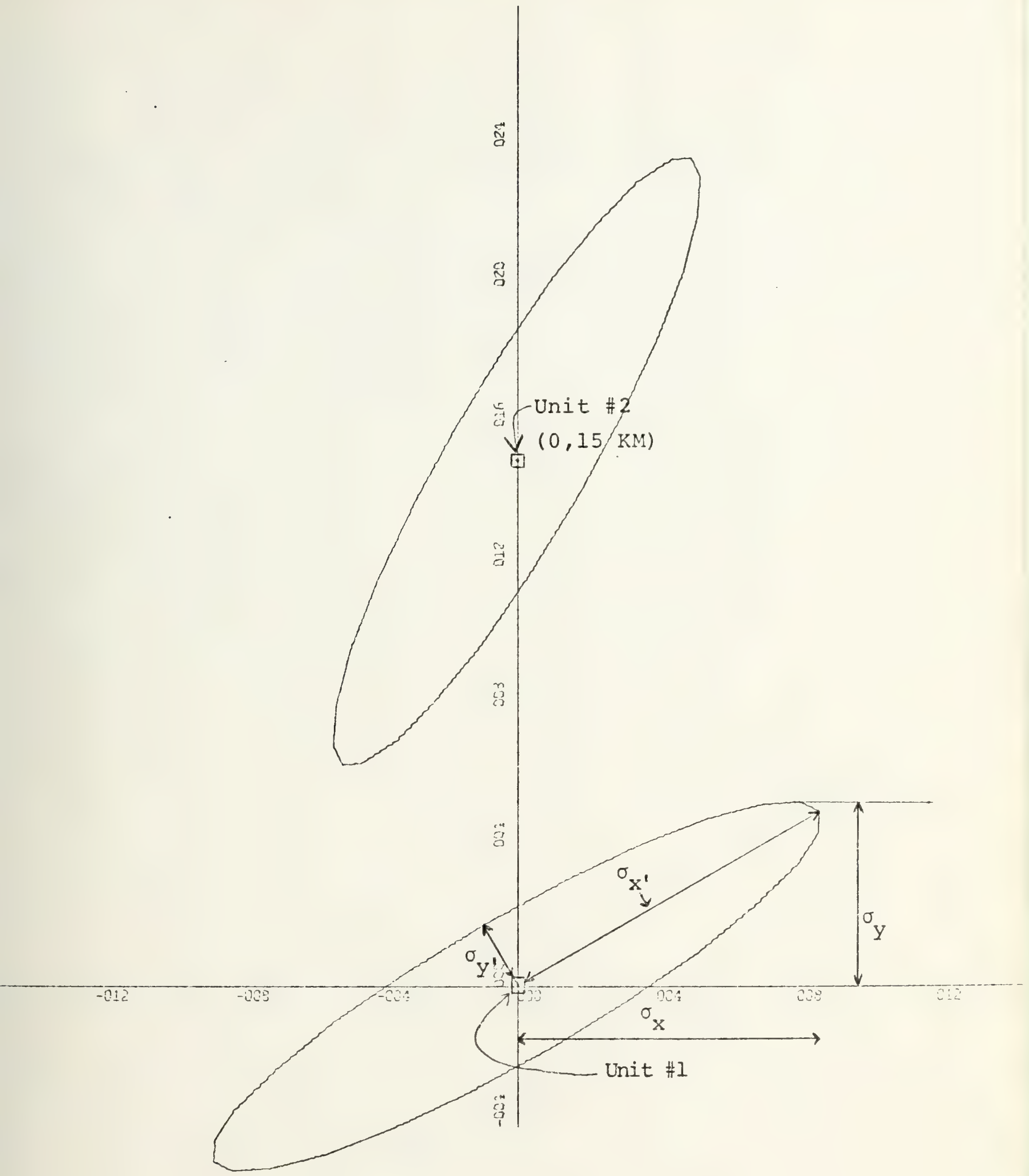

FIGURE 1. Error ellipsoids for two fixed units displaced along y-axis 15 KM

$$P_1 = \begin{bmatrix} 28 & 41.52 \\ 41.52 & 76 \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_x^{\ 2} & \sigma_x\sigma_y \\ \sigma_x\sigma_y & \sigma_y^{\ 2} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 76 & 41.52 \\ 41.52 & 28 \end{bmatrix}$$

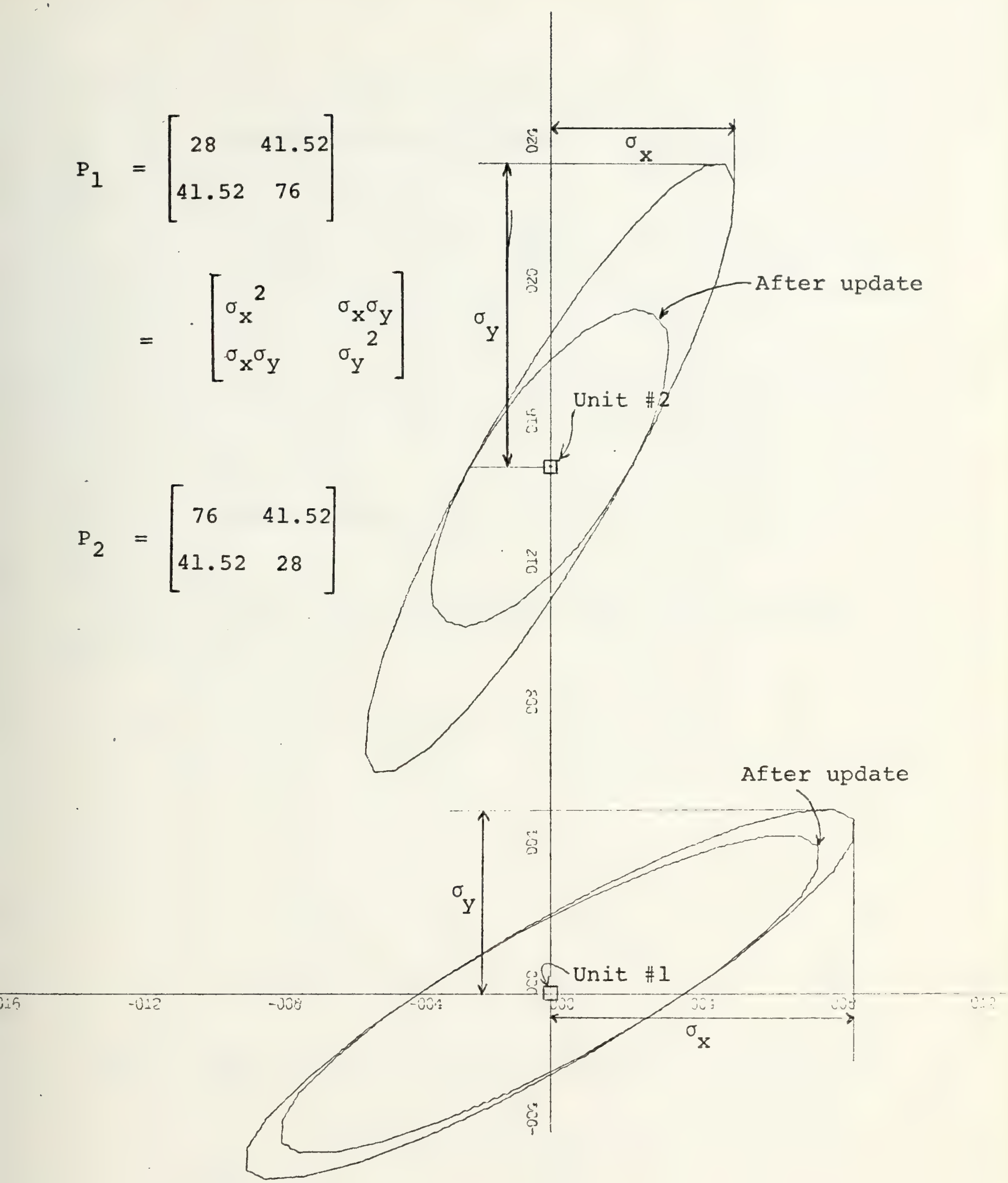

FIGURE 2. Projection of error ellipsoids to their x-y components and updated error ellipsoids after each unit ranges on the other

Thus the processing of an error free measurement between two units provides a better statistical description of both units along the range line.

This discussion considered fixed units with no velocity and only in the XY plane. Central to the equations employed in the Kalman filter are equations calculating the gains and the error covariance matrix. By merely adding the component of the variance of the ranging unit along the range line to the measurement noise in the calculation of the Kalman gains, the Kalman filter reduces the covariance matrix of the update unit properly.

The uncoupled error matrices and their directions are (from eqs. (22)-(24))

$$P_1' = \begin{bmatrix} 4 & 0 \\ 0 & 100 \end{bmatrix}, \qquad \theta_1 = -30°$$

and

$$P_2' = \begin{bmatrix} 100 & 0 \\ 0 & 4 \end{bmatrix}, \qquad \theta_2 = +30°$$

After update the resulting covariances are (see Fig. 2)

$$P_1(\text{new}) = \begin{bmatrix} 11.42 & 11.18 \\ 11.18 & 20.46 \end{bmatrix}$$

and

$$P_2(\text{new}) = \begin{bmatrix} 59.42 & 30.34 \\ 30.34 & 20.46 \end{bmatrix}$$

Their uncoupled equivalents

$$P_1'(\text{new}) = \begin{bmatrix} 3.89 & 0 \\ 0 & 28 \end{bmatrix} , \quad \theta_1(\text{new}) = -34°$$

and

$$P_2'(\text{new}) = \begin{bmatrix} 7.6 & 0 \\ 0 & 3.89 \end{bmatrix} , \quad \theta_2(\text{new}) = 28.6°$$

After ten updating operations

$$P_1 \ (\text{new}) = \begin{bmatrix} 5.33 & 0.02 \\ 0.02 & 0.04 \end{bmatrix} ,$$

$$P_2'(\text{new}) = \begin{bmatrix} 14.52 & 0.06 \\ .06 & 0.04 \end{bmatrix}$$

and their uncoupled equivalents

$$P_1'(\text{new}) = \begin{bmatrix} 5.34 & 0 \\ 0 & .08 \end{bmatrix} , \quad \theta_1'(\text{new}) = 0.47°$$

$$P_2' = \begin{bmatrix} 14.6 & 0 \\ 0 & .079 \end{bmatrix} \quad , \quad \theta_2'(\text{new}) = 0.46°$$

In Fig. 3 we see the influence of the sequential updating of the units, one upon the other over ten observations.

Note that the successive ranging in the y-direction effectively reduced the error variance in y and its projection in x due to the xy covariance. This was accomplished by assuming the measurement noise consists solely of the y component of the error variance of the other unit ranging upon it.

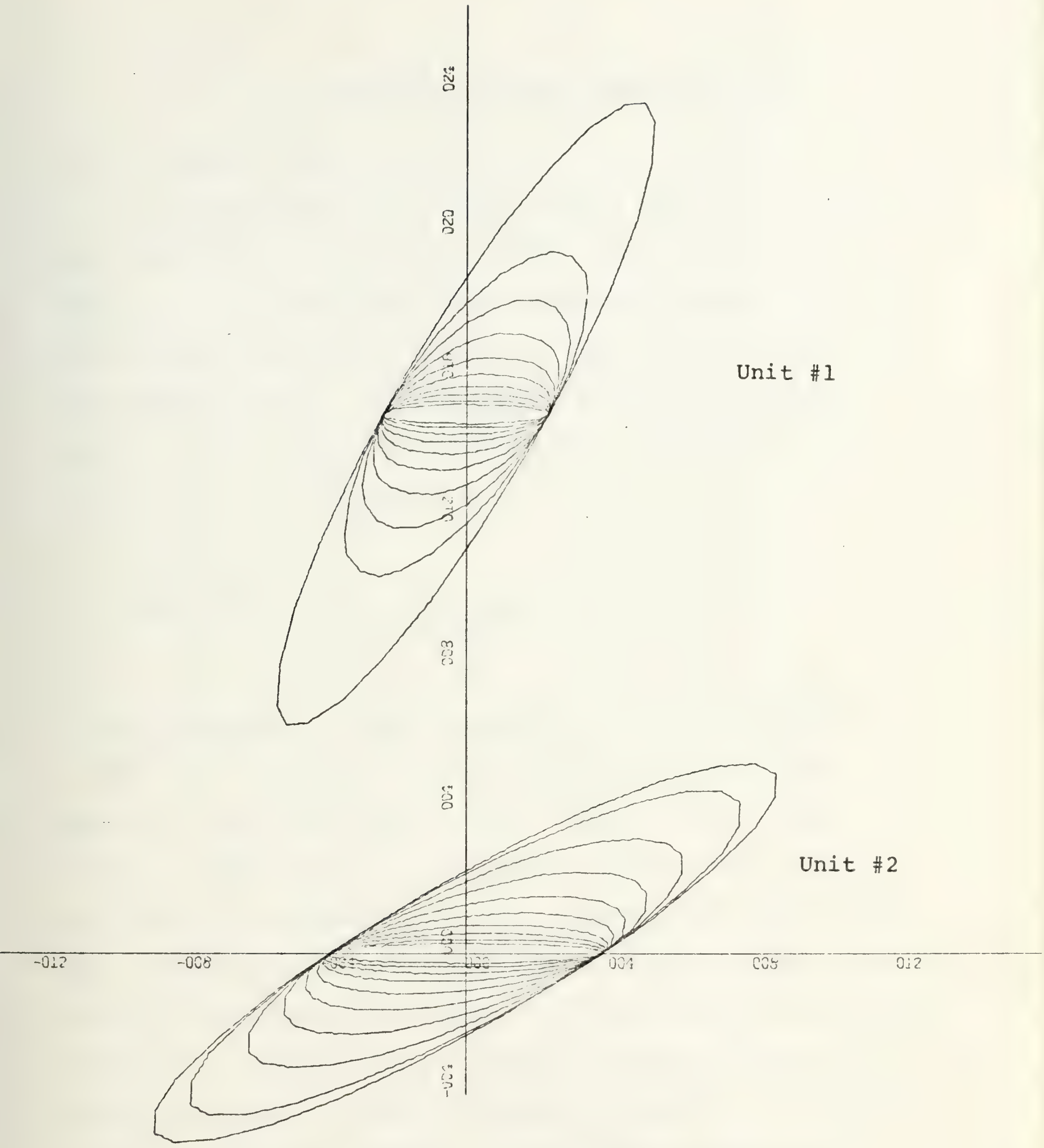

FIGURE 3. Sequential updating of the units, one upon the other, over ten observations

IV. TRACKING AND MANEUVERING TARGETS

A. THE CHOICE OF Q

One possible approach in designing a filter to accomodate maneuvers is to increase the random forcing excitation covariance, Q, until the filter adequately tracks the contemplated maneuvers. Normally this is done in a tracking simulation program. The running filter error residual, $(\underline{x}(k) - \hat{\underline{x}}(k/k))$ appears to be a better indicator of effectiveness than either mean error

$$\text{Mean Error} = \frac{1}{k} \sum_{j=1}^{k} (\underline{x}(j) - \hat{\underline{x}}(j/j))$$

or for that matter, error variance. Filter lag or bias as is best seen in the sign of the error, is lost in the squaring process for the variance. Also, the mean and variance error entail an accumulation of all error from start up to the present time. In a simulation, a maneuver may often not occur until well into a tracking run, and its effect may not appear as markedly in the mean error and error variance. A Monte Carlo simulation is often an effective design tool here. However, unless considerable engineering insight is used the resulting filter has been designed to handle the zero mean maneuver. That is, the filter will be designed against an ensemble of maneuvering tracks the mean of which is no maneuver at all. It appears that a more

successful procedure is to design against a highly probable worst case maneuver. The resulting filter should then be able to handle all possible contingencies, which is a necessary ingredient in most military systems.

The price paid in having a rather high Q in a filter design is an increased degradation in filter performance due to the measurement noise variance, R. In this application the measurement noise is minimal. The important thing is to attempt range measurements along the axis of maximum filter error variance as discussed in Section III.

B. SIMULATION RESULTS

An aircraft flying at Mach 1 follows a turn of radius 10 KM. Ranging unit 1 is located at the origin. Unit 2 is placed at (10 KM, 10 KM). We have the below filter parameters for Figs. 4 through 11 are

$$\Phi = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} ,$$

$$\Gamma = \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix}$$

and the initial conditions of

$$P(1/0) = \begin{bmatrix} 10^{-4} & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix}$$

and

$$\underline{X}(1) = X(1/0) = \begin{bmatrix} 0 \\ 0.333 \text{ KM/S} \\ 10 \text{ KM} \\ 0 \end{bmatrix}$$

In Fig. 4 we have the first attempt at the algorithm for choosing the ranging unit nearest the line generated from the major axis of error in the error ellipsoid of the tracker. It ranged solely on unit 2 and failed to follow the turn. Similarly in Fig. 5, the filter ranged on unit 2 for the first half of the flight and on unit 1 for the second half.

The evolution of the design of the filter is presented here so that the errors in the design process may help to give insight into the filter's performance.

In Fig. 6 we see the beginning of proper functioning of the ranging algorithm (except on the last 20 observations). Note also that the covariance of excitation, Q, is insufficient to follow the maneuver as is indicated by the bias

in the filter track from the true track. Here we have

$$\dot{q}_{11}/R = 2.5 \times 10^{-4}$$

from

$$R = 0.1$$

and

$$Q = \begin{bmatrix} .25\text{x}10^{-4} & .5\text{x}10^{-4} & 0 & 0 \\ .5\text{x}10^{-4} & 1\text{x}10^{-4} & 0 & 0 \\ 0 & 0 & .25\text{x}10^{-4} & .5\text{x}10^{-4} \\ 0 & 0 & .5\text{x}10^{-4} & 1\text{x}10^{-4} \end{bmatrix}$$

In Fig. 7 the algorithm appears to be functioning properly and there is sufficient Q for the filter to follow the turn. Here we have

$$q_{11}/R = 0.25 \quad .$$

In Fig. 8 the filter was tested for its response to increasing the measurement noise (from $\sigma_V = 0.01$ KM to $\sigma_V = 0.1$ KM). It tracked well. However in Fig. 9, when the noise was again increased ($\sigma_V = 1$ KM) the track was lost immediately. This gives one a ball-park indication of filter performance as the ranging observations are degraded.

Figure 10 shows the filter performance with the error ellipsoids superimposed at every fifth observation. It indicates a short malfunction of the switching algorithm when the track crosses the x-axis. This was corrected in Fig. 11. The error ellipsoids are expanded to twenty-five times their true value in order that they may be seen.

In the appendix we have listed the computer simulation program with comments regarding its operation.

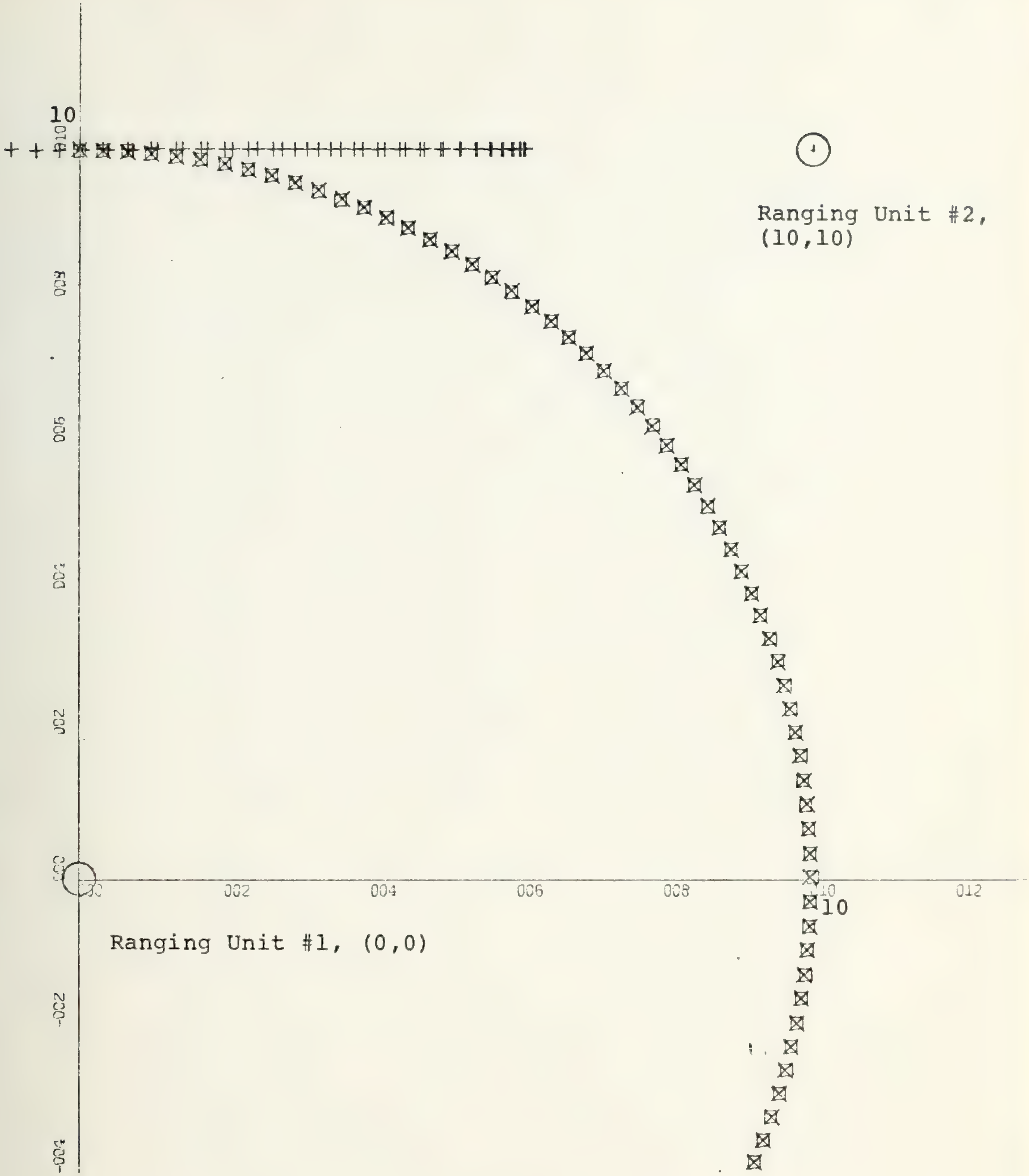

FIGURE 4. Aircraft in circular track at Mach 1 and radius 10 KM. Filter ranging only on unit #2.
x = true track, O = noise track, + = filter track

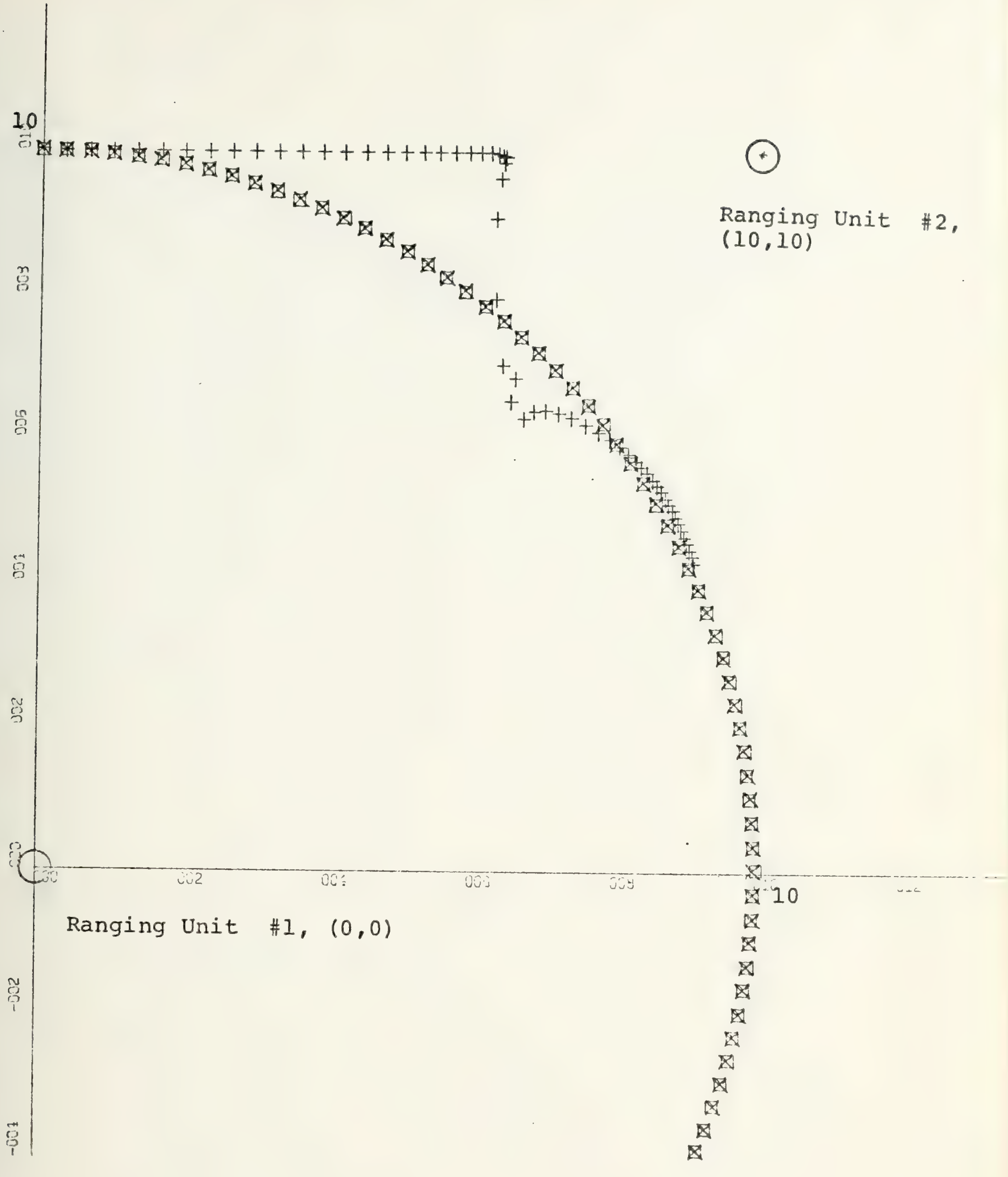

FIGURE 5. Aircraft in circular track at Mach 1 and radius 10 KM. Filter ranging on unit #2 and unit #1 during second half.
x = true track, ▢ = noise track, + = filter track

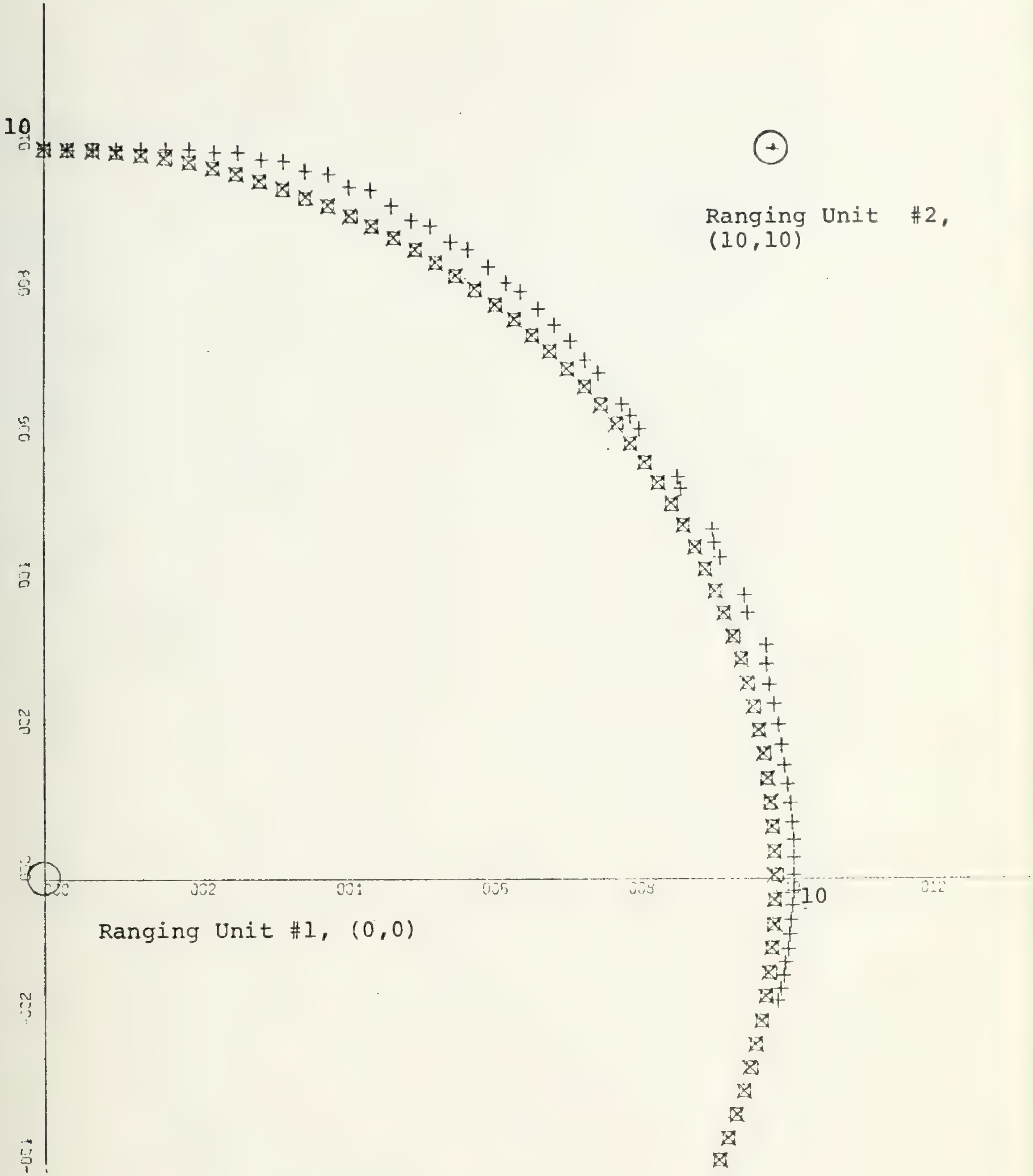

FIGURE 6. Aircraft in circular track at Mach 1 and radius 10 KM. Algorithm for choosing which ranging unit working poorly in last third of flight (unit #1 ranges solely on last 20 observations. A too low Q is indicated by the track bias.
x = true track, o = noise track, + = filter track

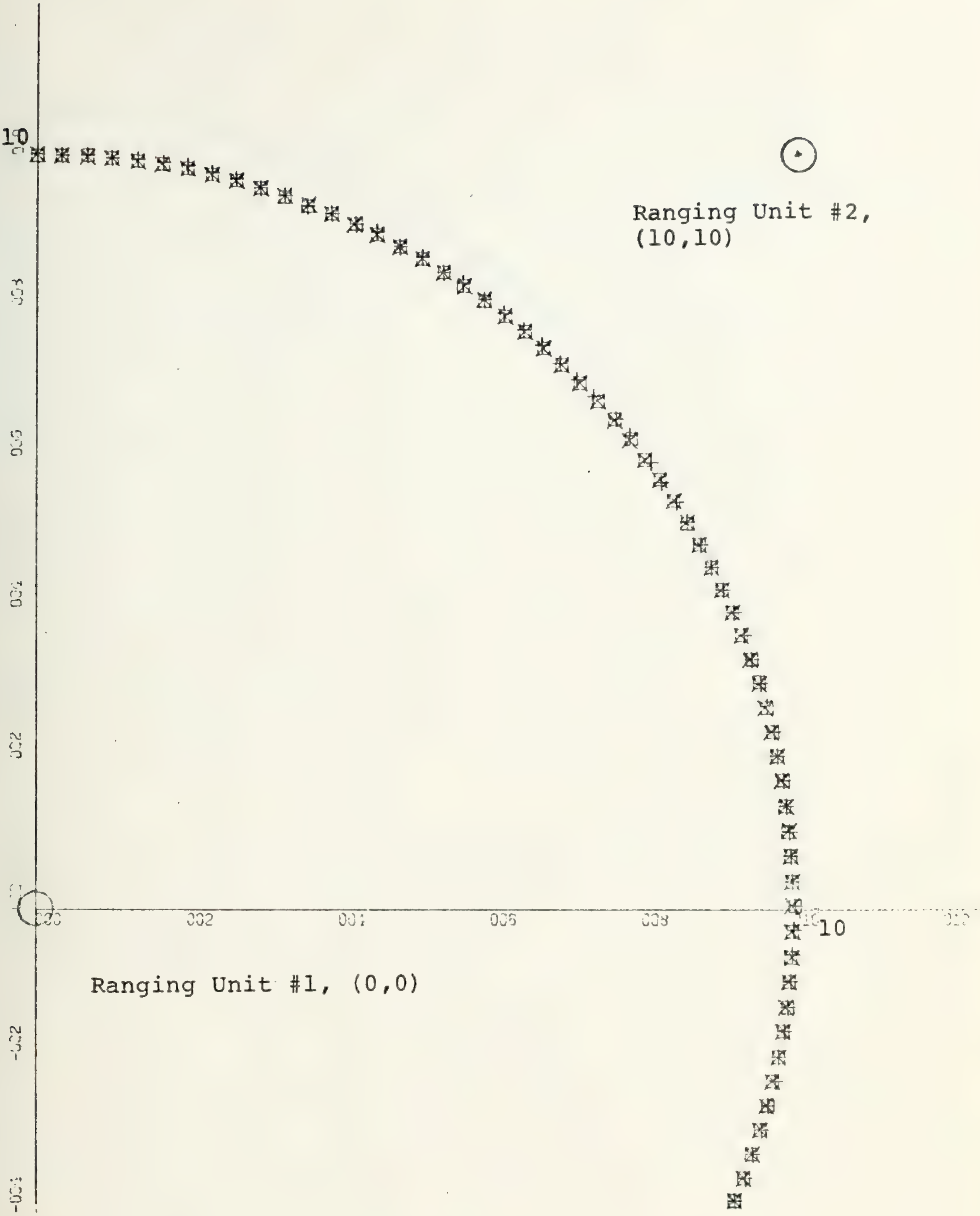

FIGURE 7. Aircraft in circular track at Mach 1 and radius 10 KM. Algorithm for choosing which ranging unit working. Q sufficient to allow filter to adequately track the turning maneuver. Measurement noise = 10 M.
x = true track, O = noise track, + = filter track

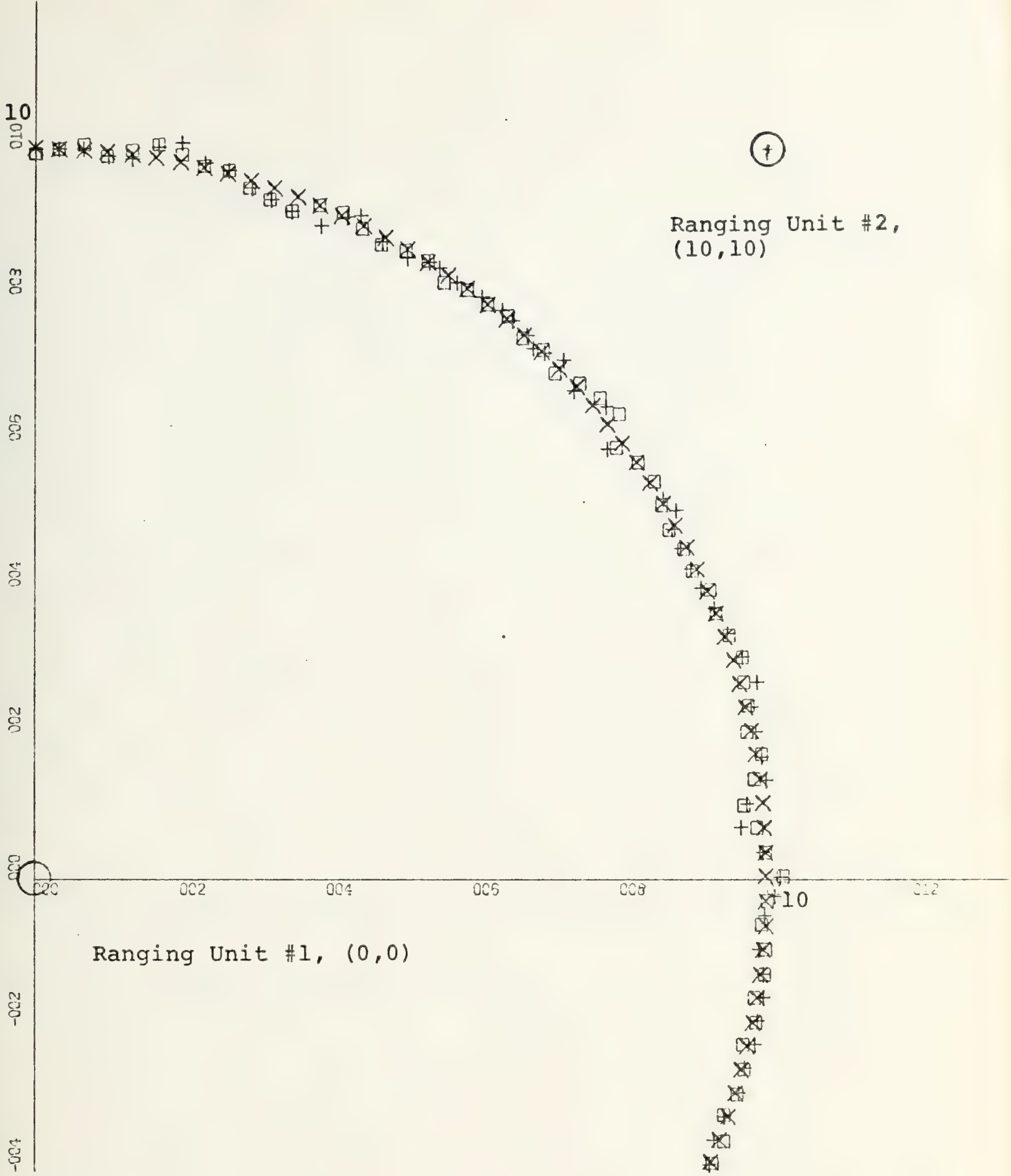

FIGURE 8. Aircraft in circular track at Mach 1 and radius 10 KM. Algorithm for choosing which ranging unit working. Q sufficient to allow filter to adequately track the turning maneuver. Measurement noise = 100 M.
x = true track, ▫ = noise track, + = filter track

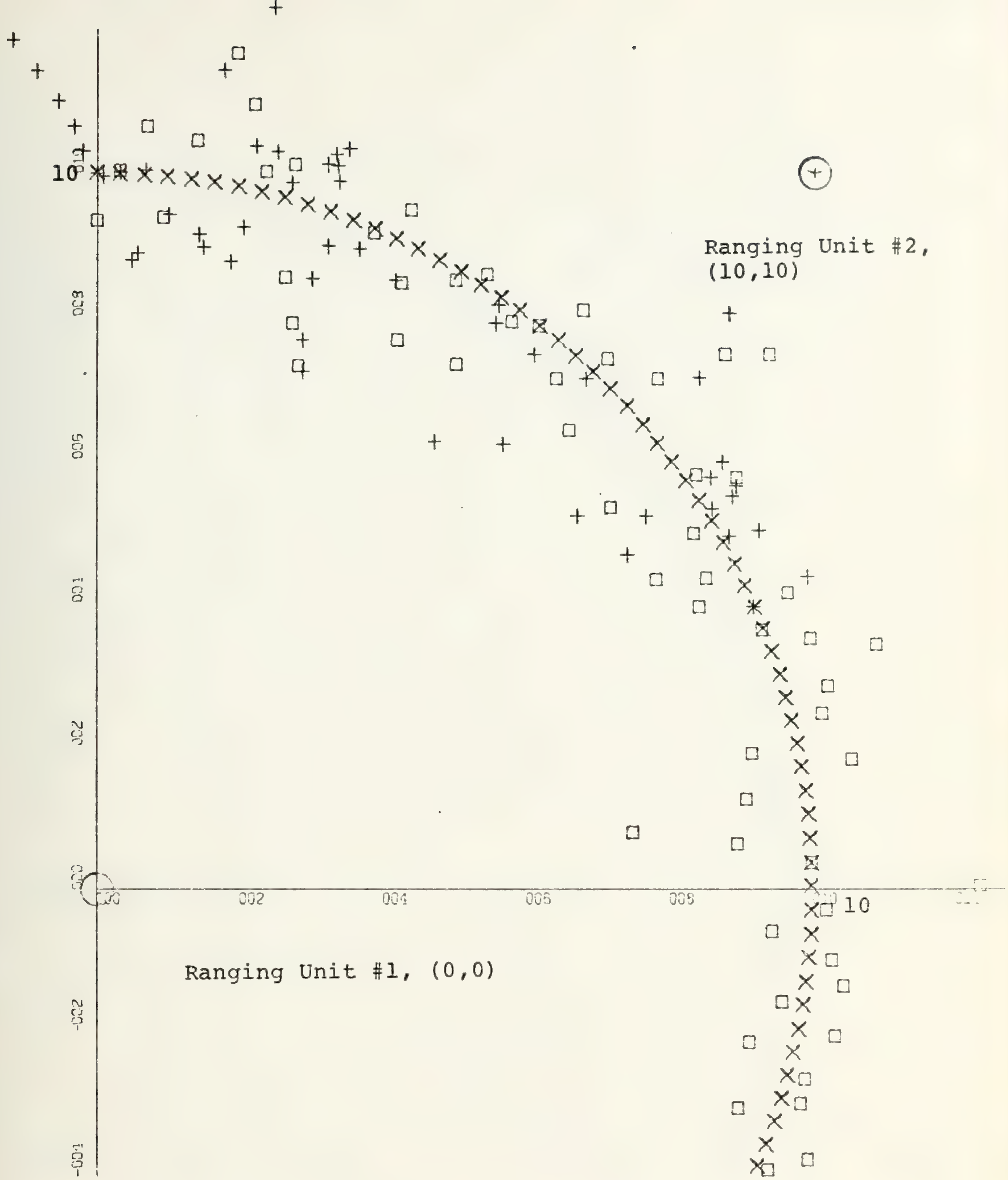

FIGURE 9. Aircraft in circular track at Mach 1 and radius 10 KM. Algorithm for choosing which ranging unit working. Q sufficient to allow filter to adequately track the turning maneuver. Filter drops track. Measurement noise = 1 KM.
x = true track, □ = noise track, + = filter track

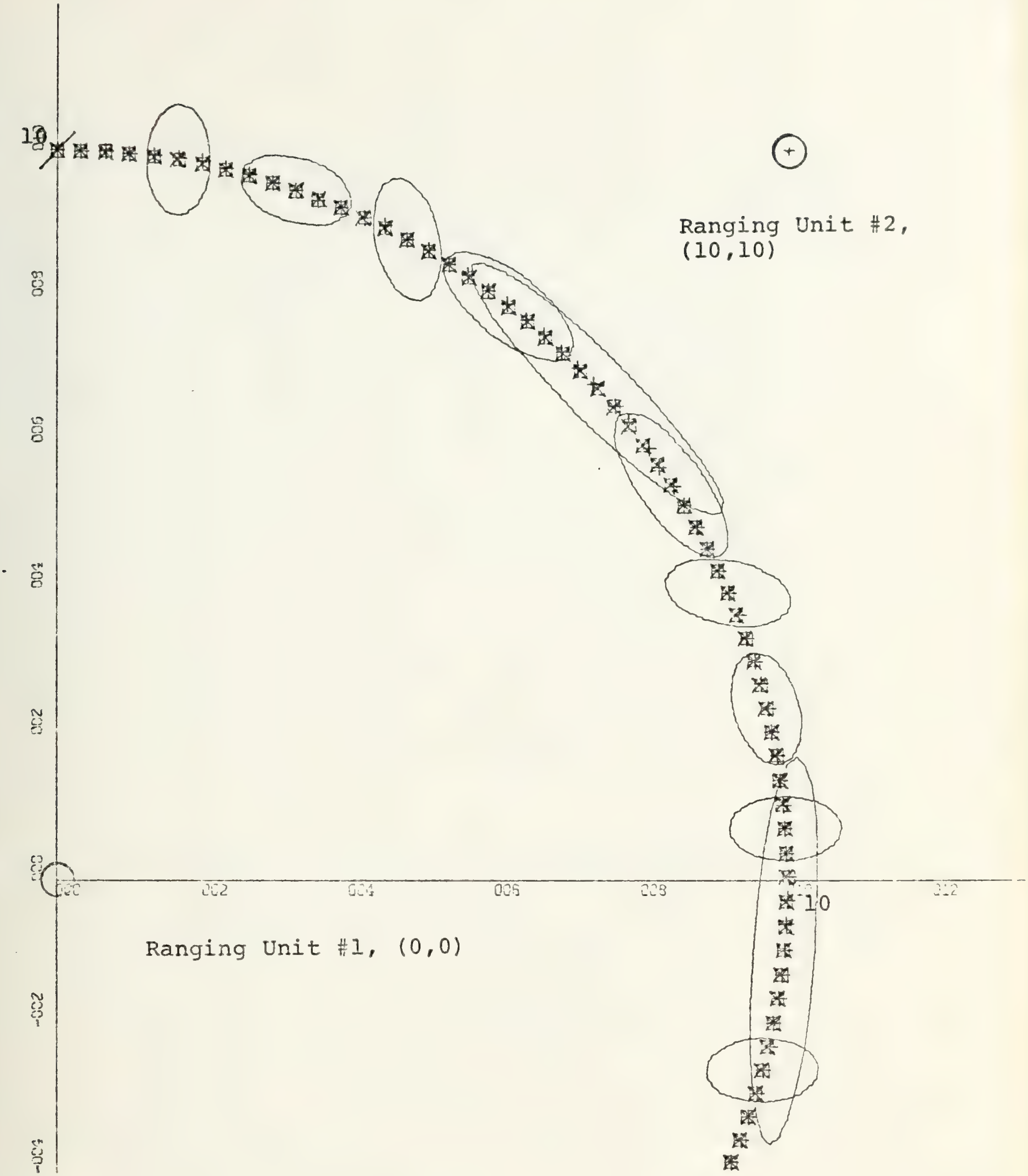

FIGURE 10. Aircraft in circular track at Mach 1 and radius of 10 KM. Error ellipsoids presented on every fifth observation. Measurement noise = 10 M. Switching algorithm fails on x-axis. Ellipses 25 times true scale.
x = true track, □ = noise track, + = filter track

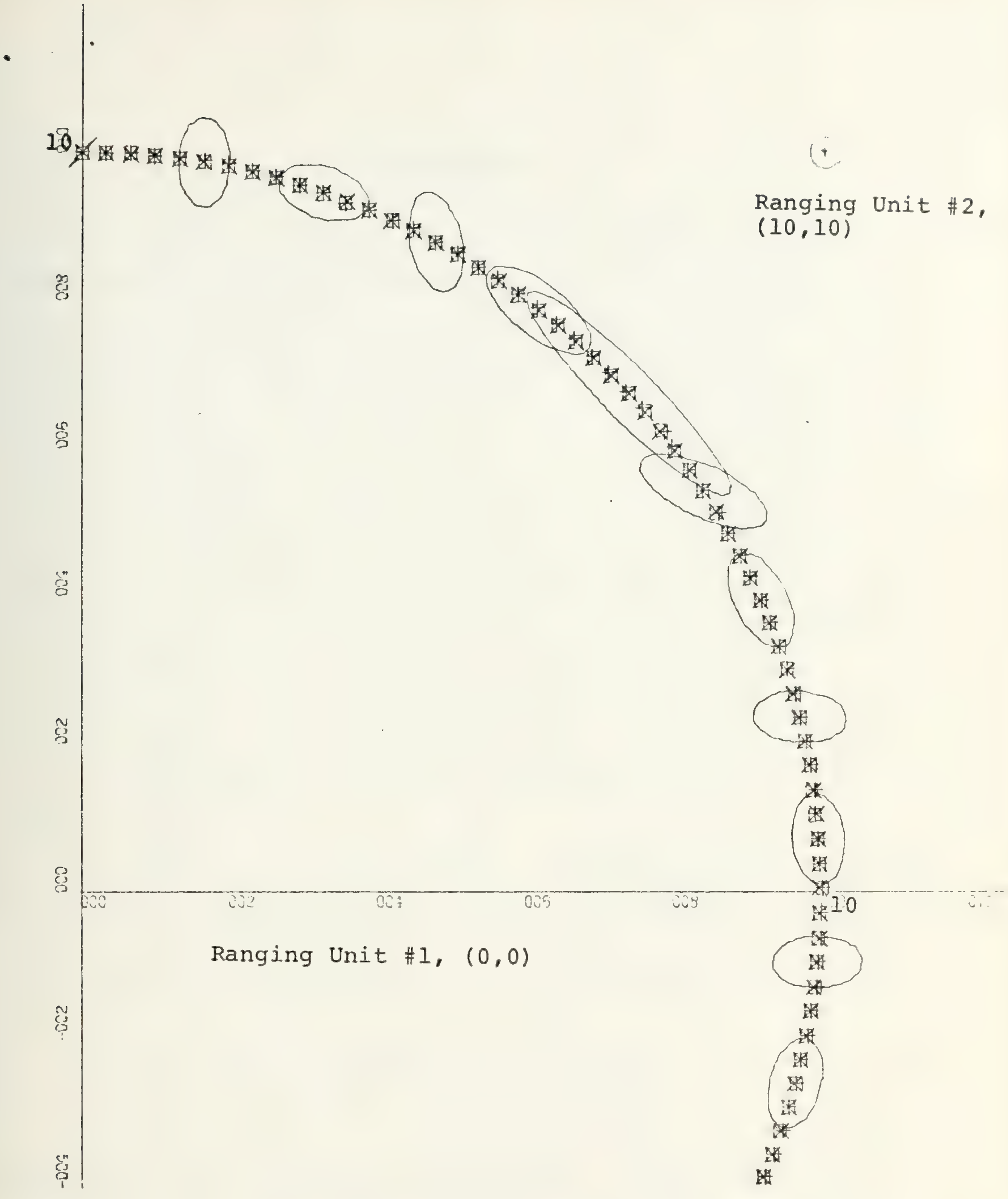

FIGURE 11. Aircraft in circular track at Mach 1 and radius of 10 KM. Switching algorithm corrected. Ellipses 20 times true scale.
x = true track, □ = noise track, + = filter track

V. CONCLUSIONS

The evolution of this filter design appears to have some merit. The maneuvering target can be tracked handily with adequate choice of Q. The filter operated well in all but impossible ranging noise situations. The real system has range noise on the order of ten meters. The filter tracked well with a range noise sigma of one hundred meters.

The algorithm for choosing which unit to link to for ranging is finally operating satisfactorily. In fact, this may be the real contribution of this work. Another possible benefit is the use of graphical presentation of the error ellipsoids in filter evaluation. They were most essential in the algorithm for choosing ranging links and in the Q evaluation.

Still to be accomplished are: an initialization algorithm, a spurious data gating scheme and a track deletion logic. The latter two items may be dispatched readily by letting

$$\text{Gate}(k) = 3\sqrt{P_{11}(k/k-1) + R_{11}(k)}$$

and if a unit receives no range update. Within the gate for, say, six update sequence times, then reinitialize and start over.

The rotation and reduction of the error ellipsoids (i.e. the filter error covariance) was most instructive and gave much insight into the performance of the filter.

FILTER FOR STATIC UNITS TRACKING

```
//   EXEC  FORTCLGP,REGION.GO=150K
//FORT.SYSIN   DD *
      REAL*4LA(8)/  'G11 ','G12 ','G22 ','G42 ','PK11','PK22','PK33',
     *'PK44'/
      DIMENSION HI(12,12),Q(12,12),H(12,12 ),R(12,12),G(12,12),
     &PHIT(12,12),PHI(12,12),XY(800),                     G12(800),
     &G22(800),           PKK(12,12),PKKM1(12,12),G11(800)
      DIMENSION PINV(12,12)
      DIMENSION IREAD(10),IWRITE(10)
      DIMENSION DEL(12,12),A(12,12),B(12,12),D1(12,12),D2(12,12)
      DIMENSIONAA(12,12),YY(12),Z(12)
      DIMENSION DELT(12,12)
      DIMENSION G21(800)
      DIMENSION G31(800),G42(800)
      DIMENSION C(12,12),D(12,12),DD(12,12)
      DIMENSION XP(50),YP(60)
      INTEGER*4  ITB(12)/12*0/
      REAL*4     RTB(28)/28*0.0/
      EQUIVALENCE (TITLE,RTB(5))
      REAL*8 TITLE(12)/'X = TRUE,  + = FILTER,  SQUARE = NOISY'/
C
C
C         THIS PROGRAM COMPUTES THE FOLLOWING KALMAN FILTER EQUATIONS
C
C                                                   -1
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)
C
C
C     P(K/K) = (I-G(K)*H)*P(K/K-1)
C
C
C     P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q
C
C
C     Q(I,J) DEFINES THE COVARIANCE OF THE PER SAMPLE RANDOM GAUSSIAN
C        EXCITATION OF THE PROCESS.
C
C     R(I,J) DEFINES THE RANDOM (GAUSSIAN) MEASUREMENT NOISE COVARIANCE
C            WHICH IS ADDED TO THE OBSERVABLE SIGNALS.
C
C     HI(I,J) IS THE IDENTITY MATRIX.
C
C     II=K(THE DISCRETE POINT IN TIME-THE STAGE OF THE PROCESS)
C
C
C     PKK(I,J) = P(K/K),(COV ERROR AT K GIVEN K SAMPLES)
C
C     PKKM1(I,J)=P(K/K-1),(COV ERROR AT K GIVEN K-1 SAMPLES)
```

```
C     N= NUMBER OF ROWS,    M= NO. OF COL.. OBS.., ND AND MD ARE THE DIMENSIONS
C     NN=NUMBER OF ITERATIONS OF FILTER
      LD=12
      READ(5,50)N,M,ND,MD,LD,NN,DT,IREAD,IWRITE
   50 FORMAT(6I5,1F10.4/10A4,10A4)
      WRITE(6,7777)
 7777 FORMAT(1H1)
      WRITE(6,51)N,M,ND,MD,LD,NN,DT
   51 FORMAT(2X,2HN=,I5,5X,2HM=,I5,5X,3HND=,I5,5X,3HMD=,I5,   5X,3HLD=,
     1I5,5X,3HNN=,I5,3HDT=,F10.4)
      CALL MREAD(R,N,N,ND,MD,IREAD)
      WRITE(6,53)
   53 FORMAT(//12H MATRIX R    /)
      CALL MWRITE(R,N,N,ND,MD,IWRITE)
      CALL MREAD(Q,N,N,ND,MD,IREAD)
      WRITE(6,54)
      CALL MWRITE(Q,N,N,ND,MD,IWRITE)
   54 FORMAT(//12H MATRIX Q    /)
      CALL MREAD(PKKM1,N,N,ND,MD,IREAD)
      WRITE(6,55)
   55 FORMAT(//13H MATRIX PKKM1/)
      CALL MWRITE(PKKM1,N,N,ND,MD,IWRITE)
C     XDOT2(0)=V*57.3*DT(SEC)/R*3600
      CALL MREAD(PHI,N,N,ND,MD,IREAD)
C     CALL PHIDEL(DT,N,M,A,B,PHI,DEL,D1,D2,ND,MD,LD)
   58 FORMAT(//13H MATRIX PHI  /)
      WRITE(6,58)
      CALL MWRITE(PHI,N,N,ND,MD,IWRITE)
      CALL MREAD(H,N,N,ND,MD,IREAD)
      WRITE(6,59)
   59 FORMAT(//13H MATRIX H    /)
      CALL MWRITE(H  ,N,N,ND,MD,IWRITE)
      CALL MREAD(HI    ,N,N,ND,MD,IREAD)
      WRITE(6,60)
   60 FORMAT(//13H MATRIX HI   /)
      CALL MWRITE(HI ,N,N,ND,MD,IWRITE)
      CALL PROD(HI,HI,N,N,N,D,ND,MD,LD)
      WRITE(6,7777)
      T=0.0
      L=1
      ST=0.0
      MM=0
      CLOCK=ITIME(0)*0.01
      WRITE(6,52)
   52 FORMAT(    //12H MATRIX G      /)
      ITB(1)=1
      ITB(8)=2
      ITB(9)=1
```

```
      ITB(10)=2
      ITB(11)=4
      RTB(1)=4
      RTB(2)=4
      DO 1000 II=1,NN
      THE1=0.5*ATAN(2.*PKKM1(1,2)/(PKKM1(1,1)-PKKM1(2,2)))
      THE2=0.5*ATAN(2.*PKKM1(3,4)/(PKKM1(3,3)-PKKM1(4,4)))
      SIG2X=( PKKM1(1,1)+PKKM1(2,2))/2.+PKKM1(1,2)/SIN(2.*THE1)
      SIG2Y=( PKKM1(1,1)+PKKM1(2,2))/2.-PKKM1(1,2)/SIN(2.*THE1)
      SIG2X2=(PKKM1(3,3)+PKKM1(4,4))/2.+PKKM1(3,4)/SIN(2.*THE2)
      SIG2Y2=(PKKM1(3,3)+PKKM1(4,4))/2.-PKKM1(3,4)/SIN(2.*THE2)
      SX=(SIG2X)**.5
      SY=(SIG2Y)**.5
      PT=3.14159265/12.
      CT=COS( THE1)
      ST=SIN( THE1)
      DO 1 I=1,25
      XI=I
      XP(I)=SX*COS( PT*XI)*CT-SY*SIN(PT*XI)*ST
    1 YP(I)=SX*COS( PT*XI)*ST+SY*SIN(PT*XI)*CT+ 15.
      CALL DRAWP(25,XP,YP,ITB,RTB)
      SX=(SIG2X2)**.5
      SY=(SIG2Y2)**.5
      CT=COS( THE2)
      ST=SIN( THE2)
      DO 7 I=1,25
      XI=I
      XP(I)=SX*COS( PT*XI)*CT-SY*SIN(PT*XI)*ST
    7 YP(I)=SX*COS( PT*XI)*ST+SY*SIN(PT*XI)*CT
      ITB(1)=2
      IF(II.LT.NN) GO TO 2
      ITB(1)=3
    2 CONTINUE
      CALL DRAWP(25,XP,YP,ITB,RTB)
      DERAD=360./(3.14159265*2.)
      THE1=DERAD*THE1
      THE2=DERAD*THE2
      WRITE (6,146) THE1,SIG2X,SIG2Y,THE2,SIG2X2,SIG2Y2
  146 FORMAT (9(2X,1PE12.5),/)
      R(1,1)=PKKM1(4,4)
      R(2,2)=PKKM1(2,2)
      CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HI,ND,MD,LD)
      ST=II/L
      IF(T-ST    )20,20,21
   20 MM=MM+1
      G11(MM)=G(1,1)
      G12(MM)=G(1,2)
      G21(MM)=G(2,1)
```

```
      G22(MM)=G(2,2)
      XY(MM)=MM-1
      WRITE(6,5)II
    5 FORMAT(/20X,5HTIME=,I5)
      CALL PROD(HI,PKKM1,N,N,N,D1,ND,MD,LD)
C   NOTE THAT THIS RECIP SR DESTROYS THE INPUT  MATRIX
      CALL RECIP(N,0.0000001,D1    ,PINV ,KER,MD)
      CALLTRANS(PHI,N,N,PHIT,ND,ND)
      CALLPROD(PKK,PHIT ,N,N,N,A,ND,MD,LD)
      CALLPROD(A,PINV,N,N,N,C,ND,MD,LD)
      CALLPROD(D,C  ,N,N,N,DD,ND,MD,LD)
      CALLPROD(DD,HI ,N,N,N,D,ND,MD,LD)
C     WRITE(6,557)
  557 FORMAT(//13H MATRIX D     )
C     CALL MWRITE(C     ,N,N,ND,MD,IWRITE)
C     CALL MWRITE(A     ,N,N,ND,MD,IWRITE)
C     CALL MWRITE(D     ,N,N,ND,MD,IWRITE)
      WRITE(6,555)
  555 FORMAT(//13H MATRIX PKK  )
      CALL MWRITE(PKK   ,N,N,ND,MD,IWRITE)
      WRITE(6,55)
      CALL MWRITE(PKKM1,N,N,ND,MD,IWRITE)
      WRITE(6,556)
  556 FORMAT(//13H MATRIX PINV )
      CALL MWRITE(PINV  ,N,N,ND,MD,IWRITE)
      WRITE(6,99)
   99 FORMAT(//13H MATRIX G     /)
      CALL MWRITE(G  ,N,N,ND,MD,IWRITE)
   21 CONTINUE
      T=T+DT
 1000 CONTINUE
      MC=0
      CALL SSPLOT(XY,G11,MM,MC,TITLE,0   ,XMIN,XMAX,YMIN,YMAX)
      CALL SSPLOT(XY,G21,MM,MC,TITLE,0   ,XMIN,XMAX,YMIN,YMAX)
      END
      SUBROUTINE SSPLOT(X,Y,N,MC,TITLE,ISC,XMIN,XMAX,YMIN,YMAX)
      DIMENSION X(1),Y(1),GRID(33,81),XS(9),YS(9),TITLE(16)
      INTEGER*4GRID,BLANK/' '/,DOT/'.'/,PLUS/'+'/,XCHAR(4)/'+','*','#',
     *'&'/
      IF(MC-1)1,1,15
    1 NC=4
      IF(ISC)13,13,14
   13 XMAX=-1.0E+20
      XMIN=-XMAX
      YMAX=XMAX
      YMIN=-XMAX
      DO 11 I=1,N
      IF(X(I)-XMAX)3,3,2
```

```
 2 XMAX=X(I)
   YXMAX=Y(I)
 3 IF(X(I)-XMIN)4,4,5
 4 XMIN=X(I)
   YXMIN=Y(I)
 5 IF(Y(I)-YMAX)7,7,6
 6 YMAX=Y(I)
   XYMAX=X(I)
 7 IF(Y(I)-YMIN)8,8,11
 8 YMIN=Y(I)
   XYMIN=X(I)
11 CONTINUE
14 IERR=0
15 IF(MC-1)32,32,21
21 NC=NC+4
   DO 31 I=1,N
   IF(X(I)-XMAX)23,23,22
22 X(I)=XMAX
   IERR=IERR+1
   GO TO 25
23 IF(X(I)-XMIN)24,25,25
24 X(I)=XMIN
   IERR=IERR+1
25 IF(Y(I)-YMAX)27,27,26
26 Y(I)=YMAX
   IERR=IERR+1
   GC TO'31
27 IF(Y(I)-YMIN)28,31,31
28 Y(I)=YMIN
   IERR=IERR+1
31 CONTINUE
32 IF(MC-2)33,71,71
33 XR=XMAX-XMIN
   YR=YMAX-YMIN
   DO 41 K=1,81
   DO 41 I=1,33
41 GRID(I,K)=BLANK
   XTEST=XMAX*XMIN
   YTEST=YMAX*YMIN
   IF(XTEST)51,51,61
51 IYAXIS=80.0*(-XMIN)/XR+1.5
   DO 52 I=1,33
52 GRID(I,IYAXIS)=DOT
61 IF(YTEST)62,62,71
62 IXAXIS=32.0*YMAX/YR+1.5
   DO 63 J=1,81
63 GRID(IXAXIS,J)=DOT
71 CONTINUE
```

```
      IF(MC-1)72,72,73
   72 JSET=0
   73 JSET=JSET+1
      IF(JSET-4)75,74,74
   74 JSET=1
   75 DO 91 I=1,N
      IPTY=32.0*(YMAX-Y(I))/YR+1.5
      IPTX=80.0*(X(I)-XMIN)/XR+1.5
      IF(IPTY-33)81,81,89
   81 IF(IPTX-81)82,82,89
   82 IF(IPTX)89,89,83
   83 IF(IPTY)89,89,84
   84 GRID(IPTY,IPTX)=XCHAR(JSET)
      GO TO 91
   89 IERR=IERR+1
   91 CONTINUE
      IF(MC-1)111,111,101
  101 IF(MC-2)103,102,103
  102 RETURN
  103 GO TO 151
  111 XINCR=XR/8.0
      YINCR=YR/8.0
      XS(1)=XMIN
      YS(1)=YMAX
      DO 121 I=2,9
  121 XS(I)=XS(I-1)+XINCR
      DO 122 I=2,9
  122 YS(I)=YS(I-1)-YINCR
      IF(MC-1)151,102,131
  131 IF(MC-2)102,102,151
  151 PRINT 80
      PRINT 10,(XS(I),I=1,9)
      II=1
      I=0
      DO 201 IK=1,33
      IF(I)161,161,162
  161 PRINT 20,YS(II),(GRID(IK,IX),IX=1,81),YS(II)
      II=II+1
      GO TO 165
  162 PRINT 30,(GRID(IK,IX),IX=1,81)
  165 I=I+1
      IF(I- 4)201,171,171
  171 I=0
  201 CONTINUE
      PRINT 40,(XS(I),I=1,9)
      PRINT 90,(TITLE(I),I=1,NC)
      IF(IERR)211,211,202
  202 PRINT 50,IERR
```

```
211 IF(ISC)212,212,102
212 PRINT 60,YMAX,XYMAX,YMIN,XYMIN
    PRINT 70,XMAX,YXMAX,XMIN,YXMIN
    RETURN
 10 FORMAT(11X,1PE9.2,8(1X,E9.2)/13X,2H**,8(10H+*********),3H+**)
 20 FORMAT(1X,1PE10.3,2X,1H+,1X,81A1,1X,1H+2X,E10.3)
 30 FORMAT(13X,1H*,1X,81A1,1X,1H*)
 40 FORMAT(13X,2H**,8(10H+*********),3H+**/11X,1PE9.2,8(1X,E9.2))
 50 FORMAT(1X,'NUMBER OF POINTS OUT OF RANGE=',I4)
 60 FORMAT(' MAX Y=',1PE12.4,' AT X=',E12.4,10X,'MIN Y=',E12.4,' AT X=
   *'E12.4)
 70 FORMAT(' MAX X=',1PE12.4,' AT Y=',E12.4,10X,'MIN X=',E12.4,' AT Y=
   *'E12.4)
 80 FORMAT(1H1)
 90 FORMAT(1X,4A4  ,4H=+, ,4A4  ,4H=*, ,4A4  ,4H=#, ,4A4  ,4H=&  )
    END
    SUBROUTINE PHIDEL(T,N,M,A,B,PHI,DEL,D1,D2,ND,MD,LD)
    DIMENSION A(12,12),B(12,12),PHI(12,12),DEL(12,12),TERM(12,12),
   1COR(12,12),C(12,12),D1(12,12),D2(12,12),TEIL(12,12)
    TEST=1.E-7
    F=1.
    DO 10 IR=1,N
    DO 10 IC=1,N
    PHI(IR,IC)=0.
    PHI(IR,IR)=1.
    C(IR,IC)=A(IR,IC)
    TEIL(IR,IC)=T/2.00*PHI(IR,IC)
 10 TERM(IR,IC)=T*PHI(IR,IC)
 50 DO 11 IR=1,N
    DO 11 IC=1,N
    COR(IR,IC)=T/F*C(IR,IC)
    PHI(IR,IC)=PHI(IR,IC)+COR(IR,IC)
    TEIL(IR,IC)=TEIL(IR,IC)+T/((F+1.   )*(F+2.   ))*COR(IR,IC)
 11 TERM(IR,IC)=TERM(IR,IC)+ T/(F+1.   ) * COR(IR,IC)
    DO 12 IR=1,N
    DO 12 IC=1,N
    C(IR,IC)=0.
    DO 12 K=1,N
 12 C(IR,IC)=C(IR,IC)+A(IR,K)*COR(K,IC)
    F=F+1.
    DO 13 IR=1,N
    DO 13 IC=1,N
    IF( ABS(COR(IR,IC)) .GT. TEST* ABS(PHI(IR,IC)))  GO TO 50
 13 CONTINUE
    CALL PROD(TERM,B,N,N,M,DEL,ND,MD,LD)
    CALL PROD(TEIL,B,N,N,M,D2 ,ND,MD,LD)
    DO 14 IR=1,N
    DO 14 IC=1,M
```

```
   14 D1(IR,IC)=DEL(IR,IC)-D2(IR,IC)
      RETURN
      END
C     THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND THE ERROR
C     COVARIANCE
      SUBROUTINE GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HI,ND,MD,LD)
     0DIMENSION PKK(12,12),Q(12,12),H(12,12)
     1 ,G(12,12),R(12,12),HI(12,12),HT(12,12),TEMP(12,12),TEMP2(12,12)
     2,TEMP1(12,12),PHI(12,12),PHIT(12,12),PKKM1(12,12)
C     G(K) =  P(K/K-1)*HT*(H*P(K/K-1)*HT + R)
      CALL TRANS(H,N,N,HT,ND,MD)
      CALL PROD(PKKM1,HT,N,N,N,TEMP,ND,MD,LD)
      CALL PROD(H,TEMP,N,N,N,TEMP1,ND,MD,LD)
      CALL ADD(TEMP1,R,N,N,TEMP1,ND,MD)
      CALL RECIP(M,0.0000001,TEMP1,TEMP2,KER,MD)
      IF (KER-2) 101,110,101
  110 WRITE(6,111)
  111 FORMAT (5HKER=2)
  101 CALL PROD(TEMP,TEMP2,N,N,N,G,ND,MD,LD)
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE
C     P(K/K) = (I-G(K)*H)*P(K/K-1)
      CALL PROD(G,H,N,N,N,TEMP,ND,MD,LD)
      DO 108 I=1,N
      DO 108 J=1,N
  108 TEMP(I,J)=-TEMP(I,J)
      CALL ADD (HI,TEMP,N,N,TEMP,ND,MD)
      CALL PROD(TEMP,PKKM1,N,N,N,PKK,ND,MD,LD)
C     NOTE HERE  PKKM1(I,J) = P(K/K-1)  WHERE
C     P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q
      CALL TRANS(PHI,N,N,PHIT,ND,MD)
      CALL PROD(PKK,PHIT,N,N,N,TEMP,ND,MD,LD)
      CALL PROD(PHI,TEMP,N,N,N,TEMP1,ND,MD,LD)
      CALL ADD(TEMP1,Q,N,N,PKKM1,ND,MD)
      RETURN
      END
      SUBROUTINE ADD (A,B,N,M,C,ND,MD)
      DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
      DO 152 I=1,N
      DC 152 J=1,M
  152 C(I,J) = A(I,J) + B(I,J)
      RETURN
      END
      SUBROUTINE SUB (A,B,N,M,C,ND,MD)
      DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
      DO 152 I=1,N
      DO 152 J=1,M
  152 C(I,J) = A(I,J) - B(I,J)
      RETURN
```

```
      END
      SUBROUTINE PROD (A,B,N,M,L,C,ND,MD,LD)
      DIMENSION A(ND,MD),B(MD,LD),C(ND,LD)
      DO 1 I=1,ND
      DO 1 J=1,LD
    1 C(I,J)=0.
      DO 151 I=1,N
      DO 151 J=1,L
      DO 151 K = 1,M
  151 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
      SUBROUTINE TRANS(A,N,M,C,ND,MD)
      DIMENSION A(ND,MD),C(MD,ND)
      DO 153 I = 1,N
      DO 153 J=1,M
  153 C(J,I) = A(I,J)
      RETURN
      END
      SUBROUTINE CONST(Q,A,N,M,C,ND,MD)
      DIMENSION A(ND,MD),C(ND,MD)
      IF(Q)11,10,11
   10 DO 100 I=1,N
      DO 100 J=1,M
  100 C(I,J) = 0.0
      RETURN
   11 IF(Q-1.0)13,12,13
   12 DO 120 I=1,N
      DO 120 J=1,M
  120 C(I,J) = A(I,J)
      RETURN
   13 IF(Q+1.0)15,14,15
   14 DO 140 I=1,N
      DO 140 J=1,M
  140 C(I,J) = -A(I,J)
      RETURN
   15 DO 150 I=1,N
      DO 150 J=1,M
  150 C(I,J) = Q*A(I,J)
      RETURN
      END
      SUBROUTINE   RECIP(N,EP,B,X,KER,M)
      DIMENSION A(12,12),X(M,M),B(M,M)
      CALL CONST(1.,B,N,N,A,12,12)
      DO 1 J=1,M
      DO 1 I=1,M
    1 X(I,J)=0.
      DO 2 K=1,N
```

```
  2 X(K,K)=1.
 10 DO 34 L=1,N
    KP=0
    Z=0.
    DO 12 K=L,N
    IF(Z.GE.ABS(A(K,L))) GO TO 12
 11 Z=ABS(A(K,L))
    KP=K
 12 CONTINUE
    IF(L.GE.KP) GO TO 20
 13 DO 14 J=L,N
    Z=A(L,J)
    A(L,J)=A(KP,J)
 14 A(KP,J)=Z
    DO 15 J=1,N
    Z=X(L,J)
    X(L,J)=X(KP,J)
 15 X(KP,J)=Z
 20 IF(ABS(A(L,L)).LE.EP) GO TO 50
 30 IF(L.GE.N) GO TO 34
 31 LP1=L+1
    DO 36 K=LP1,N
    IF(A(K,L).EQ.0.) GO TO 36
 32 RATIO=A(K,L)/A(L,L)
    DO 33 J=LP1,N
 33 A(K,J)=A(K,J)-RATIO*A(L,J)
    DO 35 J=1,N
 35 X(K,J)=X(K,J)-RATIO*X(L,J)
 36 CONTINUE
 34 CONTINUE
 40 DO 43 I=1,N
    I1=N+1-I
    DO 43 J=1,N
    S=0.
    IF(I1.GE.N) GO TO 43
 41 I1P1=I1+1
    DO 42 K=I1P1,N
 42 S=S+A(I1,K)*X(K,J)
 43 X(I1,J)=(X(I1,J)-S)/A(I1,I1)
    KER=1
    RETURN
 50 KER=2
    RETURN
    END
    SUBROUTINE MREAD(A,N,M,ND,MD,IREAD)
    DIMENSION A(ND,MD),IREAD(10)
    DO 10 I=1,N
 10 READ(5,20)(A(I,J),J=1,M)
```

```
   20 FORMAT(8F10.5)
      RETURN
      END
      SUBROUTINE MWRITE(A,N,M,ND,MD,IWRITE)
      DIMENSION A(ND,MD),IWRITE(10)
      DO 10 I=1,N
   10 WRITE(6,20)(I,J,A(I,J),J=1,M)
   20 FORMAT(6(3X,'(',I2,',',I2,')= ',1PE10.3))
      RETURN
      END
//GO.FT06F001 DD SPACE=(CYL,(6,1))
//GO.SYSIN DD *
    4    2   12   12   12   10 1.

0.





28.         41.52
41.52       76.
                      76.        41.52
                       41.52      28.

1.
            1.
                      1.
                                  1.
              1.
                                 1.


1.
           1.
                      1.
                                 1.
```

```
// EXEC FORTCLGP,REGION.GO=200K
//FORT.SYSIN DD *
C     THIS PROGRAM PERFORMS MONTE CARLO SIMULATION OF STATE ESTIMATORS   MCSP0000
C     OF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL       MCSP0001
C     OPTIONS AVAILABLE AS INDICATED IN THE DETAILLED COMMENTS BELOW.    MCSP0002
C     IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE        MCSP0003
C     SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL        MCSP0004
C     ARRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCORDINGLY.             MCSP0005
C                                                                        MCSP0006
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI        MCSP0007
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),        MCSP0008
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0009
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),           MCSP0010
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),       MCSP0011
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),    MCSP0012
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND            MCSP0014
      DIMENSION XP(80),YP(80)                                            MCSP0015
C                                                                        MCSP0016
C                                                                        MCSP0017
C     N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSION OF X,XHAT)           MCSP0018
C                                                                        MCSP0019
C     M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)               MCSP0020
C                                                                        MCSP0021
C     IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)           MCSP0022
C                                                                        MCSP0023
C     NSAM=NUMBER OF TIME SAMPLES                                        MCSP0024
C                                                                        MCSP0025
C     NENS=NUMBER OF MEMBERS IN ENSEMBLE                                 MCSP0026
C                                                                        MCSP0027
      READ (5,81) N,M,IN,NSAM,NENS                                       MCSP0028
C                                                                        MCSP0029
C                                                                        MCSP0030
      READ (5,82) ND                                                     MCSP0031
C     THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION MCSP0032
C     SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS     MCSP0033
C     SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.         MCSP0034
C                                                                        MCSP0035
C                                                                        MCSP0036
C     IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN                       MCSP0037
C         0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO      MCSP0038
C         1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE                MCSP0039
C                                                                        MCSP0040
C     IFLR=0 -- R IS READ IN                                             MCSP0041
C     IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE             MCSP0042
C                                                                        MCSP0043
C     IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED                  MCSP0044
C     IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED                       MCSP0045
```

```
C                                                                       MCSP0046
C     ITRK= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS          MCSP0047
C     ITRK= 1 -- ONLY ONE TRACK IS USED                                 MCSP0048
C      ITRK.NE.OOR1 -- SEVERAL TRACKS GENERATED BUT NOT FROM STD.       MCSP0049
C                      LINEAR DIFFERENCE EQUATIONS                      MCSP0050
C                                                                       MCSP0051
C     ISTAT=0 -- MEAN OF TRACK,MEAN & VARIANCE OF EST. ERROR COMPUTED   MCSP0052
C     ISTAT.NE.0 -- SAME AS ISTAT=0 BUT OFF-DIAGONAL TERMS IN COVARIANCEMCSP0053
C                                                                       MCSP0054
C                  MATRIX ARE ALSO COMPUTED                             MCSP0055
C                                                                       MCSP0056
      READ (5,83) IG,IFLR,IEST,ITRK,ISTAT,IQ,ITRO                       MCSP0057
C                                                                       MCSP0058
C     IQ=-1 -- Q IS COMPUTED ON-LINE AT EACH SAMPLE BY "QON"            MCSP0059
C         0 -- Q MATRIX IS READ IN                                      MCSP0060
C         1 -- COVARIANCE OF W READ IN AND Q IS  COMPUTED ONCE BEFORE   MCSP0061
C              MONTE CARLO BEGINS BY CALLING "QMAT"                     MCSP0062
C                                                                       MCSP0063
C     ITRO=0 -- ONLY USED IF ONE TRACK; ITRO=0 INDICATES THE            MCSP0064
C               TRACK IS READ IN (S.P.), OTHERWISE THE TRACK            MCSP0065
C               IS GENERATED BY A SINGLE CALL OF SUBROUTINE             MCSP0066
C               TRACK.                                                  MCSP0067
C                                                                       MCSP0068
C     IPRT=0 -- SOME OR ALL OUTPUT DATA IS PRINTED                      MCSP0069
C                                                                       MCSP0070
C     IPLT=0 -- SOME OR ALL OUTPUT DATA IS PLOTTED                      MCSP0071
      READ (5,84) IPRT,IPLT                                             MCSP0072
C                                                                       MCSP0073
C     IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT ARE PLOTTING INDICATORS         MCSP0074
C     IF THE INDICATOR IS EQUAL TO 1 PLOTS ARE GENERATED. THE INDICATORSMCSP0075
C     CORRESPOND TO THE FOLLOWING QUANTITIES                            MCSP0076
C     IGPLT(GAIN VS K),ITHVPL (THEORETICAL COVARIANCE MATRIX VS K),     MCSP0077
C     IMTPLT (SAMPLE MEAN OF TRACK=TRACK IF ONLY ONE TRACK),            MCSP0078
C     ISMPLT (SAMPLE MEAN OF EST. ERROR VS K), ISVPLT (SAMPLE VARIANCE  MCSP0079
C     OF EST. ERROR VS K)                                               MCSP0080
C                                                                       MCSP0082
C     IPHI=0 -- PHI MATRIX TO BE READ IN (IN DOUBLE PREC)               MCSP0083
C     IPHI.NE.0 -- PHI MATRIX NOT TO BE READ                            MCSP0084
C                                                                       MCSP0085
C     IH=0 -- H MATRIX TO BE READ IN (IN D.P.)                          MCSP0086
C                                                                       MCSP0087
C     IR=0 -- R MATRIX IS READ IN                                       MCSP0088
C                                                                       MCSP0089
C     IPKKM1=0 -- THE STARTING VALUE FOR THE MATRIX P(K/K-1) IS TO      MCSP0090
C                 BE READ IN (D.P.)                                     MCSP0091
C                                                                       MCSP0092
C     IGAM=0 -- GAMMA MATRIX TO BE READ IN (D.P.)                       MCSP0093
C                                                                       MCSP0094
```

```
C     ISIGV=0 -- STD. DEVIATIONS OF MEASUREMENT NOISE TO BE READ IN (SP)MCSP0095
C                                                                       MCSP0096
C     ISIGW=0 -- STD. DEVIATIONS OF RANDOM INPUTS TO BE READ IN (SP)    MCSP0097
C                                                                       MCSP0098
C     IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO        MCSP0099
C               BE READ IN (S.P.)                                       MCSP0100
C                                                                       MCSP0101
      READ (5,86) IPHI,IH,IR,IPKKM1,IGAM,ISIGV,ISIGW,IXHZ,IC            MCSP0107
      CALL OVFLOW                                                       MCSP0108
      IW = 6395217                                                      MCSP0109
      IV = 1936743                                                      MCSP0110
      IXZ = 135769                                                      MCSP0111
C                                                                       MCSP0112
C                                                                       MCSP0113
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS      MCSP0114
C     SPECIFIED BY THE USER'S FLAGS                                     MCSP0115
C                                                                       MCSP0116
C                                                                       MCSP0117
      WRITE (6,156)                                                     MCSP0118
      WRITE (6,87)                                                      MCSP0119
      IF (IG.EQ.0) GO TO 1                                              MCSP0120
      IF (IG.EQ.1) GO TO 2                                              MCSP0121
      WRITE (6,88)                                                      MCSP0122
      GO TO 3                                                           MCSP0123
    1 WRITE (6,89)                                                      MCSP0124
      GO TO 3                                                           MCSP0125
    2 WRITE (6,90)                                                      MCSP0126
    3 IF (IEST,EQ.0) GO TO 4
      WRITE (6,91)                                                      MCSP0128
      GO TO 5                                                           MCSP0129
    4 WRITE (6,92)                                                      MCSP0130
    5 IF (ITRK.EQ.0) GO TO 8                                            MCSP0131
      IF (ITRK.EQ.-1) GO TO 7                                           MCSP0132
      IF (ITRO.EQ.0) GO TO 6                                            MCSP0133
      WRITE (6,93)                                                      MCSP0134
      GO TO 9                                                           MCSP0135
    6 WRITE (6,94)                                                      MCSP0136
      GO TO 9                                                           MCSP0137
    7 WRITE (6,95)                                                      MCSP0138
      GO TO 9                                                           MCSP0139
    8 WRITE (6,96)                                                      MCSP0140
    9 IF (ISTAT.EQ.0) GO TO 10                                          MCSP0141
      WRITE (6,97)                                                      MCSP0142
      GO TO 11                                                          MCSP0143
   10 WRITE (6,98)                                                      MCSP0144
   11 IF (IQ.EQ.0) GO TO 13                                             MCSP0145
      IF (IQ.EQ.1) GO TO 12                                             MCSP0146
      WRITE (6,99)                                                      MCSP0147
```

```
      GO TO 14                                                          MCSP0148
   12 WRITE (6,100)                                                     MCSP0149
      GO TO 14                                                          MCSP0150
   13 WRITE (6,101)                                                     MCSP0151
   14 IF (IFLR.EQ.0) GO TO 15                                           MCSP0152
      WRITE (6,102)                                                     MCSP0153
      GO TO 16                                                          MCSP0154
   15 WRITE (6,103)                                                     MCSP0155
   16 WRITE (6,104)                                                     MCSP0156
      IF (IPHI.EQ.0) WRITE (6,105)                                      MCSP0157
      IF (IH.EQ.0) WRITE (6,106)                                        MCSP0158
      IF (IFLR.EQ.0) WRITE (6,107)                                      MCSP0159
      IF (IQ.EQ.0) GO TO 17                                             MCSP0160
      IF (IQ.EQ.-1) GO TO 18                                            MCSP0161
      WRITE (6,108)                                                     MCSP0162
      GO TO 18                                                          MCSP0163
   17 WRITE (6,109)                                                     MCSP0164
   18 IF (IGAM.EQ.0) WRITE (6,110)                                      MCSP0165
      IF (ISIGV.EQ.0) WRITE (6,111)                                     MCSP0166
      IF (ISIGW.EQ.0) WRITE (6,112)                                     MCSP0167
      IF (IXHZ.EQ.0) WRITE (6,113)                                      MCSP0168
      IF (IC.NE.1) WRITE (6,114)                                        MCSP0169
      IF (IPKKM1.EQ.0) WRITE (6,115)                                    MCSP0170
C                                                                       MCSP0171
C                                                                       MCSP0172
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE OUTPUT      MCSP0173
C     DATA CALLED FOR                                                   MCSP0174
C                                                                       MCSP0175
C                                                                       MCSP0176
      WRITE (6,116)                                                     MCSP0177
      IF (IPRT.NE.0) GO TO 19                                           MCSP0178
      WRITE (6,117)                                                     MCSP0179
      IF (IG.EQ.0) WRITE (6,118)                                        MCSP0180
      WRITE (6,119)                                                     MCSP0181
      IF (ISTAT.NE.0) WRITE (6,120)                                     MCSP0182
      GO TO 20                                                          MCSP0183
   19 WRITE (6,121)                                                     MCSP0184
   20 IF (IPLT.NE.0) GO TO 21                                           MCSP0185
      WRITE (6,122)                                                     MCSP0186
      IF (IGPLT.EQ.1) WRITE (6,123)                                     MCSP0187
      IF (ITHVPL.EQ.1) WRITE (6,124)                                    MCSP0188
      IF (IMTPLT.EQ.1) WRITE (6,125)                                    MCSP0189
      IF (ISMPLT.EQ.1) WRITE (6,126)                                    MCSP0190
      IF (ISVPLT.EQ.1) WRITE (6,127)                                    MCSP0191
      GO TO 22                                                          MCSP0192
   21 WRITE (6,128)                                                     MCSP0193
   22 CONTINUE                                                          MCSP0194
      WRITE (6,156)                                                     MCSP0195
```

```
      WRITE (6,129)
      WRITE (6,130) N,M,IN,NSAM,NENS,ND
C
C
C     THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES
C
      IF (IPHI.NE.0) GO TO 24
      CALL MREAD (PHI,N,N)
C
      DO 23 I=1,N
C
      DO 23 J=1,N
   23 PHIS(I,J) = PHI(I,J)
C
      WRITE (6,131)
      CALL MWRITE (PHI,N,N)
C
C
   24 IF (IH.NE.0) GO TO 26
      CALL MREAD (H,M,N)
C
      DO 25 I=1,M
C
      DO 25 J=1,N
   25 HS(I,J) = H(I,J)
C
      WRITE (6,132)
      CALL MWRITE (H,M,N)
C
C
   26 IF (IR.NE.0) GO TO 27
      CALL MREAD (R,M,M)
      WRITE (6,133)
      CALL MWRITE (R,M,M)
C
C
   27 IF (IQ.NE.1) GO TO 28
      CALL MREAD (COVW,IN,IN)
      WRITE (6,134)
      CALL MWRITE (COVW,IN,IN)
      GO TO 29
   28 IF (IQ.NE.0) GO TO 29
      CALL MREAD (Q,N,N)
      WRITE (6,135)
      CALL MWRITE (Q,N,N)
C
C
   29 IF (IGAM.NE.0) GO TO 31
```

```
      CALL MREAD (GAMMA,N,IN)                                           MCSP0244
C                                                                       MCSP0245
      DO 30 I=1,N                                                       MCSP0246
C                                                                       MCSP0247
      DO 30 J=1,IN                                                      MCSP0248
   30 GAMMAS(I,J) = GAMMA(I,J)                                          MCSP0249
C                                                                       MCSP0250
      WRITE (6,136)                                                     MCSP0251
      CALL MWRITE (GAMMA,N,IN)                                          MCSP0252
C                                                                       MCSP0253
C                                                                       MCSP0253
   31 IF (IPKKM1.NE.0) GO TO 32                                         MCSP0254
      CALL MREAD (PKKM1,N,N)                                            MCSP0255
      WRITE (6,137)                                                     MCSP0256
      CALL MWRITE (PKKM1,N,N)                                           MCSP0257
C                                                                       MCSP0258
C                                                                       MCSP0259
   32 IF (ISIGV.NE.0) GO TO 33                                          MCSP0260
      CALL VREAD (SIGV,M)                                               MCSP0261
      WRITE (6,138)                                                     MCSP0262
      CALL VWRITE (SIGV,M)                                              MCSP0263
C                                                                       MCSP0264
C                                                                       MCSP0265
   33 IF (ISIGW.NE.0) GO TO 34                                          MCSP0266
      CALL VREAD (SIGW,IN)                                              MCSP0267
      WRITE (6,139)                                                     MCSP0268
      CALL VWRITE (SIGW,IN)                                             MCSP0269
C                                                                       MCSP0270
C                                                                       MCSP0271
   34 IF (IXHZ.NE.0) GO TO 35                                           MCSP0272
      CALL VREAD (XHATZ,N)                                              MCSP0273
      WRITE (6,140)                                                     MCSP0274
      CALL VWRITE (XHATZ,N)                                             MCSP0275
C                                                                       MCSP0276
C                                                                       MCSP0277
   35 IF (IC.EQ.1) GO TO 36                                             MCSP0278
C     IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE MCSP0279
C     VALUE MUST BE READ IN. OTHERWISE NOT READ IN.                     MCSP0280
      CALL VREAD (XZMEAN,N)                                             MCSP0281
      WRITE (6,141)                                                     MCSP0282
      CALL VWRITE (XZMEAN,N)                                            MCSP0283
C                                                                       MCSP0284
C                                                                       MCSP0285
      CALL VREAD (SIGXZ,N)                                              MCSP0286
      WRITE (6,142)                                                     MCSP0287
      CALL VWRITE (SIGXZ,N)                                             MCSP0288
      GO TO 40                                                          MCSP0289
C                                                                       MCSP0290
                                                                        MCSP0291
```

```
C
   36 READ (5,144) (XS(I,1),I=1,N)
C     INITIAL CONDITION HAS BEEN READ
      WRITE (6,143)
      WRITE (6,146) (XS(I,1),I=1,N)
      IF (ITRK.NE.1) GO TO 40
      IF (ITRO.NE.0) GO TO 38
C
      DO 37 K=2,NSAM
   37 READ (5,144) (XS(I,K),I=1,N)
C
      GO TO 39
   38 CALL TRACK
C     IF TRACK CALLED HERE IT SHOULD BE WRITTEN TO GENERATE AND
C     STORE THE TRACK IN XS(N,K) FOR K=2(NOTE),NSAM
C
   39 WRITE (6,145)
      WRITE (6,146) (XS(I,1),I=1,N)
      WRITE (6,146) (XS(I,NSAM),I=1,N)
   40 CONTINUE
C
C
C     THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP
C
C     FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION
C
      DO 41 I=1,N
C
      DO 41 J=1,N
      EI(I,J) = 0.D0
   41 IF (I.EQ.J) EI(I,J)=1.D0
C
C     GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q
C     USING DOUBLE PRECISION ARITHMETIC
      IF (IQ.NE.1) GO TO 42
      CALL QMAT
      WRITE (6,135)
      CALL MWRITE (Q,N,N)
   42 IF (IG.NE.-1) GO TO 44
C
      DO 43 K=1,NSAM
C
      DO 43 I=1,N
   43 READ (5,144) (GKS(I,J,K),J=1,M)
C
      GO TO 47
   44 IF (IG.NE.0) GO TO 47
C
```

```
      DO 46 K=1,NSAM                                                    MCSP0340
      CALL GAIN                                                         MCSP0341
C                                                                       MCSP0342
      DO 46 I=1,N                                                       MCSP0343
C                                                                       MCSP0344
      DO 45 L=1,N                                                       MCSP0345
   45 PKKS(I,L,K) = PKK(I,L)                                            MCSP0346
C                                                                       MCSP0347
C                                                                       MCSP0348
      DO 46 J=1,M                                                       MCSP0349
   46 GKS(I,J,K) = G(I,J)                                               MCSP0350
C                                                                       MCSP0351
   47 CONTINUE                                                          MCSP0352
C     IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY              MCSP0353
C     ONCE(IG=0), THIS HAS NOW BEEN DONE                                MCSP0354
C                                                                       MCSP0355
C                                                                       MCSP0356
C     SET UP ARRAYS FOR COMPUTING STATISTICS                            MCSP0357
C                                                                       MCSP0358
      DC 48 K=1,NSAM                                                    MCSP0359
C                                                                       MCSP0360
      DO 48 J=1,N                                                       MCSP0361
      XM(J,K) = 0.                                                      MCSP0362
      ERR(J,K) = 0.                                                     MCSP0363
C                                                                       MCSP0364
      DO 48 L=1,N                                                       MCSP0365
   48 VAR(J,L,K) = 0.                                                   MCSP0366
C                                                                       MCSP0367
C                                                                       MCSP0368
C                                                                       MCSP0369
C     BEGIN MAIN ITERATION LOOP HERE                                    MCSP0370
C                                                                       MCSP0371
      DC 54 ITER=1,NENS                                                 MCSP0372
      IF (IC.EQ.1) GO TO 49                                             MCSP0373
      CALL XZERO                                                        MCSP0374
C                                                                       MCSP0375
   49 DO 50 I=1,N                                                       MCSP0376
   50 XHKKM1(I) = XHATZ(I)                                              MCSP0377
C                                                                       MCSP0378
C                                                                       MCSP0379
      DO 54 K=1,NSAM                                                    MCSP0380
C     FORM NOISY MEASUREMENT FROM TRUE STATE VALUE                      MCSP0381
C                                                                       MCSP0382
C                                                                       MCSP0383
      DO 51 I=1,N                                                       MCSP0384
   51 X(I) = XS(I,K)                                                    MCSP0385
C                                                                       MCSP0386
      CALL MEAS                                                         MCSP0387
```

```
C     GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1
      IF (IG.NE.1) GO TO 53
C
C     Q IS TO BE COMPUTED ON-LINE IF IFLQ.NE.0
      IF (IFLQ.NE.0) CALL QON
C     R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0
C
      IF (IFLR.NE.0) CALL RON
      CALL GAIN
C
      DO 52 I=1,N
C
      DO 52 J=1,M
   52 GKS(I,J,K) = G(I,J)
C
C
C     UPDATE THE STATE ESTIMATE
C
   53 CALL ESTIM
C
C     UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS
      CALL STAT
      IF (K.EQ.NSAM) GO TO 54
C
C     UPDATE TRACK BY COMPUTING X(K+1)
      IF (ITRK.NE.1) CALL TRACK
   54 CONTINUE
C
C
C     DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE
C     SIZE TO COMPUTE STATISTICS
      ENS = NENS
C
      DO 56 K=1,NSAM
C
      DO 56 J=1,N
      IF (ITRK.EQ.1) GO TO 55
      XM(J,K) = XM(J,K)/ENS
   55 ERR(J,K) = ERR(J,K)/ENS
   56 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(J,K)**2
C
      IF (ISTAT.EQ.0) GO TO 58
C
C     COMPUTE OFF-DIAGONAL TERMS IN COVARIANCE OF ESTIMATION
C     ERROR MATRIX IF ISTAT.NE.0
C
      DO 57 K=1,NSAM
C
```

```
      DO 57 L=2,N                                                        MCSP0436
      LM1 = L-1                                                          MCSP0437
C                                                                        MCSP0438
      DO 57 J=1,LM1                                                      MCSP0439
   57 VAR(L,J,K) = VAR(L,J,K)/ENS-ERR(L,K)*ERR(J,K)                      MCSP0440
C                                                                        MCSP0441
   58 CONTINUE                                                           MCSP0442
C                                                                        MCSP0443
C                                                                        MCSP0444
      IF (IPRT.NE.0) GO TO 64                                            MCSP0445
      CALL PRT
   64 IF (IPLT.NE.0) GO TO 80                                            MCSP0489
C                                                                        MCSP0490
      CALL PLT
   80 CONTINUE                                                           MCSP0552
      STOP                                                               MCSP0553
C                                                                        MCSP0554
   81 FORMAT (5(I10))                                                    MCSP0555
   82 FORMAT (I2)                                                        MCSP0556
   83 FORMAT (7(I10))                                                    MCSP0557
   84 FORMAT (2(I5))                                                     MCSP0558
   85 FORMAT (5(I10))
   86 FORMAT (9I5)                                                       MCSP0560
   87 FORMAT (20X,'DESCRIPTION OF RUN',//)                               MCSP0561
   88 FORMAT (10X,'GAINS COMPUTED OFF-LINE AND READ IN',/)               MCSP0562
   89 FORMAT (10X,'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE CMCSP0563
     1ARLO',/)                                                           MCSP0564
   90 FORMAT (10X,'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',/)        MCSP0565
   91 FORMAT (10X,'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILTMCSP0566
     1ER',/)                                                             MCSP0567
   92 FORMAT (10X,'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',MCSP0568
     1/)                                                                 MCSP0569
   93 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUTMCSP0570
     1INE TRACK',/)                                                      MCSP0571
   94 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS READ IN',/)          MCSP0572
   95 FORMAT (10X,'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEAMCSP0573
     1R DIFFERENCE EQS.',/)                                              MCSP0574
   96 FORMAT (10X,'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIFMCSP0575
     1FERENCE EQS.',/)                                                   MCSP0576
   97 FORMAT (10X,'MEAN OF TRACK, MEAN OF EST. ERROR AND COVARIANCE OF EMCSP0577
     1ST. ERROR ARE COMPUTED',/)                                         MCSP0578
   98 FORMAT (10X,'MEAN OF TRACK, MEAN AND VARRIANCES OF EST. ERROR ARE MCSP0579
     1COMPUTED',/)                                                       MCSP0580
   99 FORMAT (10X,'THE Q MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "QMCSP0581
     1ON"',/)                                                            MCSP0582
  100 FORMAT (10X,'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "MCSP0583
     1QMAT" BEFORE STARTING MONTE CARLO',/)                              MCSP0584
  101 FORMAT (10X,'THE Q MATRIX IS READ IN',/)                           MCSP0585
```

```
102 FORMAT (10X,'R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON"',/)     MCSP0586
103 FORMAT (10X,'R IS READ IN',/)                                      MCSP0587
104 FORMAT (////,20X,'INPUT DATA CALLED FOR',//)                       MCSP0588
105 FORMAT (10X,'PHI MATRIX',/)                                        MCSP0589
106 FORMAT (10X,'H MATRIX',/)                                          MCSP0590
107 FORMAT (10X,'R MATRIX',/)                                          MCSP0591
108 FORMAT (10X,'COVARIANCE OF W',/)                                   MCSP0592
109 FORMAT (10X,'Q MATRIX',/)                                          MCSP0593
110 FORMAT (10X,'GAMMA MATRIX',/)                                      MCSP0594
111 FORMAT (10X,'STANDARD DEVIATIONS OF MEASUREMENT NOISE',/)          MCSP0595
112 FORMAT (10X,'STANDARD DEVIATIONS OF INPUT FORCING W',/)            MCSP0596
113 FORMAT (10X,'XHAT(0/-1)',/)                                        MCSP0597
114 FORMAT (10X,'MEANS AND VARIANCES OF X(0)',/)                       MCSP0598
115 FORMAT (10X,'P(0/-1)',/)                                           MCSP0599
116 FORMAT (////,20X,'OUTPUT CALLED FOR',//)                           MCSP0600
117 FORMAT (10X,'PRINTED OUTPUT OF THE FOLLOWING DATA',/)              MCSP0601
118 FORMAT (15X,'GAIN MATRICES AND THEORETICAL COVARIANCE OF EST. ERROMCSP0602
   1R MATRICES',/)                                                     MCSP0603
119 FORMAT (15X,'SAMPLE MEANS OF TRACK AND ESTIMATION ERROR,SAMPLE VARMCSP0604
   1IANCES OF EST. ERROR',/)                                           MCSP0605
120 FORMAT (15X,'COVARIANCE OF ESTIMATION ERROR MATRIX',/)             MCSP0606
121 FORMAT (10X,'NO PRINTED OUTPUT CALLED FOR',//)                     MCSP0607
122 FORMAT (10X,'THE FOLLOWING PLOTS ARE CALLED FOR',/)                MCSP0608
123 FORMAT (15X,'G(K) VS. K',/)                                        MCSP0609
124 FORMAT (15X,'P(K/K) THEORETICAL VS. K',/)                          MCSP0610
125 FORMAT (15X,'MEAN OF TRACK VS. K',/)                               MCSP0611
126 FORMAT (15X,'SAMPLE MEANS OF ESTIMATION ERROR VS. K',/)            MCSP0612
127 FORMAT (15X,'SAMPLE VARIANCES OF ESTIMATION ERROR VS. K',/)        MCSP0613
128 FORMAT (10X,'NO PLOTS CALLED FOR',/)                               MCSP0614
129 FORMAT (20X,'INPUT DATA',///)                                      MCSP0615
130 FORMAT (4X,'N=',I2,4X,'M=',I2,4X'IN=',I2,4X'NSAM=',I3,4X,'NENS=',IMCSP0616
   15,4X'ND=',I2,//)                                                   MCSP0617
131 FORMAT (//,10X,'THE PHI MATRIX IS',/)                              MCSP0618
132 FORMAT (//,10X,'THE H MATRIX IS',/)                                MCSP0619
133 FORMAT (//,10X,'THE R MATRIX IS',/)                                MCSP0620
134 FORMAT (//,10X,'THE COVARIANCE OF W MATRIX IS',/)                  MCSP0621
135 FORMAT (//,10X,'THE Q MATRIX IS',/)                                MCSP0622
136 FORMAT (//,10X,'THE GAMMA MATRIX IS',/)                            MCSP0623
137 FORMAT (//,10X,'THE MATRIX P(0/-1) IS',/)                          MCSP0624
138 FORMAT (//,10X,'THE STD. DEVIATIONS OF MEASUREMENT NOISE ARE',/)   MCSP0625
139 FORMAT (//,10X,'THE STD. DEVIATIONS OF INPUT FORCING W ARE',/)     MCSP0626
140 FORMAT (//,10X,'THE VECTOR XHAT(0/-1) IS',/)                       MCSP0627
141 FORMAT (//,10X,'THE MEAN OF THE VECTOR X(0) IS',/)                 MCSP0628
142 FORMAT (//,10X,'THE STANDARD DEVIATIONS OF THE VECTOR X(0) ARE',/)MCSP0629
143 FORMAT (//,10X,'THE INITIAL STATE IS',/)                           MCSP0630
144 FORMAT (4F20.0)                                                    MCSP0631
145 FORMAT (//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO BMCSP0632
   1E USED ARE',/)                                                     MCSP0633
```

```
  146 FORMAT (9(2X,1PE12.5),/)
  156 FORMAT ('1')
      END                                                               MCSP0654
      SUBROUTINE QMAT                                                   MCSP0764
C                                                                       MCSP0765
C     THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION           MCSP0766
C                                                                       MCSP0767
C        Q=GAMMA* E(W*WT) * GAMMAT                                      MCSP0768
C                                                                       MCSP0769
C     DOUBLE PRECISION ARITHMETIC IS USED                               MCSP0770
C                                                                       MCSP0771
C                                                                       MCSP0772
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0773
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0774
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0775
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0776
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0777
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0778
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0780
C                                                                       MCSP0781
C                                                                       MCSP0782
      CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)                               MCSP0783
      CALL TRANS (GAMMA,N,IN,TEMP1)                                     MCSP0784
      CALL PROD (TEMP,TEMP1,N,IN,N,Q)                                   MCSP0785
      RETURN                                                            MCSP0786
      END                                                               MCSP0787
      SUBROUTINE QON                                                    MCSP0788
C                                                                       MCSP0789
C     IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE             MCSP0790
C     IN THIS SUBROUTINE                                                MCSP0791
C                                                                       MCSP0792
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0793
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0794
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0795
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0796
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0797
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0798
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0800
C                                                                       MCSP0801
C     THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST           MCSP0802
C     BE INSERTED HERE BY THE USER                                      MCSP0803
C                                                                       MCSP0804
      RETURN                                                            MCSP0805
      END                                                               MCSP0806
      SUBROUTINE RON                                                    MCSP0807
C                                                                       MCSP0808
```

```
C     IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE             MCSP0809
C     IN THIS SUBROUTINE                                                MCSP0810
C                                                                       MCSP0811
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0812
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0813
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0814
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0815
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0816
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0817
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0819
C                                                                       MCSP0820
C     THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST           MCSP0821
C     BE INSERTED HERE BY THE USER                                      MCSP0822
C                                                                       MCSP0823
      RETURN                                                            MCSP0824
      END                                                               MCSP0825
      SUBROUTINE STAT                                                   MCSP0826
C     THIS SUBROUTINE  COMPUTES RUNNING SUMS USED IN DETERMINING THE    MCSP0827
C     SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT  MCSP0828
C     OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF     MCSP0829
C     TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION        MCSP0830
C     ERROR. IF(ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF MCSP0831
C     ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.                        MCSP0832
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0833
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0834
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0835
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0836
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0837
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0838
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0840
      DIMENSION EXH(3)                                                  MCSP0841
      IF (ITRK.NE.1) GO TO 2                                            MCSP0842
      IF (ITER.NE.1) GO TO 4                                            MCSP0843
C                                                                       MCSP0844
      DO 1 J=1,N                                                        MCSP0845
    1 XM(J,K) = XS(J,K)                                                 MCSP0846
C                                                                       MCSP0847
      GO TO 4                                                           MCSP0848
    2 CONTINUE                                                          MCSP0849
C                                                                       MCSP0850
      DO 3 J=1,N                                                        MCSP0851
    3 XM(J,K) = XM(J,K)+XS(J,K)                                         MCSP0852
C                                                                       MCSP0853
    4 CONTINUE                                                          MCSP0854
C                                                                       MCSP0855
      DO 5 J=1,N                                                        MCSP0856
```

```
      EXH(J) = XHKK(J)-XS(J,K)                                          MCSP0857
      ERR(J,K) = ERR(J,K)+EXH(J)                                        MCSP0858
    5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2                                 MCSP0859
C                                                                       MCSP0860
      IF (ISTAT.EQ.0) RETURN                                            MCSP0861
C                                                                       MCSP0862
      DO 6 L=2,N                                                        MCSP0863
      LM1 = L-1                                                         MCSP0864
C                                                                       MCSP0865
      DO 6 J=1,LM1                                                      MCSP0866
    6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)                             MCSP0867
C                                                                       MCSP0868
      RETURN                                                            MCSP0869
      END                                                               MCSP0870
      SUBROUTINE XZERO                                                  MCSP0934
C     THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL   MCSP0935
C     RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE     MCSP0936
C     HAS COMPONENTS THAT ARE INDEPENDENT                               MCSP0937
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0938
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0939
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0940
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0941
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0942
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0943
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0945
      CALL SNORM (IXZ,X,N)                                              MCSP0946
C                                                                       MCSP0947
      DO 1 I=1,N                                                        MCSP0948
    1 XS(I,1) = SIGXZ(I)*X(I)+XZMEAN(I)                                 MCSP0949
C                                                                       MCSP0950
      RETURN                                                            MCSP0951
      END                                                               MCSP0952
      SUBROUTINE ADD (A,B,N,M,C)                                        MCSP0953
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE        MCSP0954
C     RESULT IN C                                                       MCSP0955
      REAL*8 A,B,C                                                      MCSP0956
      DIMENSION A(4,4),B(4,4),C(4,4)                                    MCSP0957
C                                                                       MCSP0958
      DO 1 I=1,N                                                        MCSP0959
C                                                                       MCSP0960
      DO 1 J=1,M                                                        MCSP0961
    1 C(I,J) = A(I,J)+B(I,J)                                            MCSP0962
C                                                                       MCSP0963
      RETURN                                                            MCSP0964
      END                                                               MCSP0965
      SUBROUTINE MREAD (A,N,M)                                          MCSP0966
C     8D10.5. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN    MCSP0968
```

```
C     THIS SUBROUTINE READS AN NXM MATRIX A ACCORCING TO THE FORMAT     MCSP0967
C     THE ENTRIES IN THE SECOND ROW, AND SO ON.                         MCSP0969
      REAL*8 A                                                          MCSP0970
      DIMENSION A(4,4)                                                  MCSP0971
C                                                                       MCSP0972
      DO 1 I=1,N                                                        MCSP0973
    1 READ (5,2) (A(I,J),J=1,M)                                         MCSP0974
C                                                                       MCSP0975
      RETURN                                                            MCSP0976
C                                                                       MCSP0977
    2 FORMAT (8F10.0)                                                   MCSP0978
      END                                                               MCSP0979
      SUBROUTINE MWRITE (A,N,M)                                         MCSP0980
C     THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A            MCSP0981
      REAL*8 A                                                          MCSP0982
      DIMENSION A(4,4)                                                  MCSP0983
C                                                                       MCSP0984
      DO 1 I=1,N                                                        MCSP0985
    1 WRITE (6,2) (A(I,J),J=1,M)                                        MCSP0986
C                                                                       MCSP0987
      RETURN                                                            MCSP0988
C                                                                       MCSP0989
    2 FORMAT (9(2X,1PE12.5))                                            MCSP0990
      END                                                               MCSP0991
      SUBROUTINE PROD (A,B,N,M,L,C)                                     MCSP0992
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE     MCSP0993
C     RESULT IN C                                                       MCSP0994
C     A = NXM,    B = MXL,    C = NXL                                   MCSP0995
      REAL*8 A,B,C,T                                                    MCSP0996
      DIMENSION A(4,4),B(4,4),C(4,4),T(4,4)                             MCSP0997
C                                                                       MCSP0998
      DO 1 I=1,N                                                        MCSP0999
C                                                                       MCSP1000
      DO 1 J=1,L                                                        MCSP1001
    1 T(I,J) = 0.0                                                      MCSP1002
C                                                                       MCSP1003
C                                                                       MCSP1004
      DO 2 I=1,N                                                        MCSP1005
C                                                                       MCSP1006
      DO 2 J=1,L                                                        MCSP1007
C                                                                       MCSP1008
      DO 2 K=1,M                                                        MCSP1009
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)                                     MCSP1010
C                                                                       MCSP1011
C                                                                       MCSP1012
      DO 3 I=1,N                                                        MCSP1013
C                                                                       MCSP1014
      DO 3 J=1,L                                                        MCSP1015
```

```
      3 C(I,J) = T(I,J)                                                 MCSP1016
C                                                                       MCSP1017
        RETURN                                                          MCSP1018
        END                                                             MCSP1019
        SUBROUTINE SUB (A,B,N,M,C)                                      MCSP1020
C       THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX  MCSP1021
C       A AND STORES THE RESULT IN C                                    MCSP1022
        REAL*8 A,B,C                                                    MCSP1023
        DIMENSION A(4,4),B(4,4),C(4,4)                                  MCSP1024
C                                                                       MCSP1025
        DO 1 I=1,N                                                      MCSP1026
C                                                                       MCSP1027
        DO 1 J=1,M                                                      MCSP1028
      1 C(I,J) = A(I,J)-B(I,J)                                          MCSP1029
C                                                                       MCSP1030
        RETURN                                                          MCSP1031
        END                                                             MCSP1032
        SUBROUTINE TRANS (A,N,M,C)                                      MCSP1033
C       THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE     MCSP1034
C       RESULT IN C                                                     MCSP1035
C       A = NXM,   C = MXN                                              MCSP1036
        REAL*8 A,C                                                      MCSP1037
        DIMENSION A(4,4),C(4,4)                                         MCSP1038
C                                                                       MCSP1039
        DO 1 I=1,N                                                      MCSP1040
C                                                                       MCSP1041
        DO 1 J=1,M                                                      MCSP1042
      1 C(J,I) = A(I,J)                                                 MCSP1043
C                                                                       MCSP1044
        RETURN                                                          MCSP1045
        END                                                             MCSP1046
        SUBROUTINE VADD (X,Y,N,Z)                                       MCSP1047
C       THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND         MCSP1048
C       Y AND STORES THE RESULT IN THE N-VECTOR Z                       MCSP1049
C                                                                       MCSP1050
        REAL*4 X(4),Y(4),Z(4)                                           MCSP1051
C                                                                       MCSP1052
        DO 1 I=1,N                                                      MCSP1053
      1 Z(I) = X(I)+Y(I)                                                MCSP1054
C                                                                       MCSP1055
        RETURN                                                          MCSP1056
        END                                                             MCSP1057
        SUBROUTINE VPROD (A,X,M,N,Y)                                    MCSP1058
C       THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX          MCSP1059
C       A AND THE N-VECTOR X AND STORES THE RESULT IN THE               MCSP1060
C       M-VECTOR Y                                                      MCSP1061
C                                                                       MCSP1062
C                                                                       MCSP1063
```

```
      REAL*4 A(4,4),X(4),Y(4),T(4)                                      MCSP1064
C                                                                       MCSP1065
      DC 1 I=1,M                                                        MCSP1066
      T(I) = 0.DO                                                       MCSP1067
C                                                                       MCSP1068
      DC 1 J=1,N                                                        MCSP1069
    1 T(I) = T(I)+A(I,J)*X(J)                                           MCSP1070
C                                                                       MCSP1071
C                                                                       MCSP1072
      DC 2 I=1,M                                                        MCSP1073
    2 Y(I) = T(I)                                                       MCSP1074
C                                                                       MCSP1075
      RETURN                                                            MCSP1076
      END                                                               MCSP1077
      SUBROUTINE VREAD (V,N)                                            MCSP1078
C                                                                       MCSP1079
C     THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V             MCSP1080
C                                                                       MCSP1081
      DIMENSION V(4)                                                    MCSP1082
      READ (5,1) (V(I),I=1,N)                                           MCSP1083
      RETURN
    1 FCRMAT (8F10.0)
      END
      SUBROUTINE VSUB (X,Y,N,Z)
      REAL*4 X(4),Y(4),Z(4)
      DC 1 I=1,N
    1 Z(I) = X(I)-Y(I)
      RETURN
      END
      SUBROUTINE VWRITE (V,N)
      DIMENSION V(4)
      WRITE (6,1) (V(I),I=1,N)
      RETURN
    1 FCRMAT (9(2X,1PE12.5))
      END
      SUBROUTINE TRACK                                                  MCSP0871
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DCNE IN THIS SUBROUTINE MCSP0872
C     IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED          MCSP0873
C     FRCM THE STANDARD LINEAR DIFFERENCE EQUATION                      MCSP0874
C                                                                       MCSP0875
C     X(K+1)=PHI*X(K)+GAMMA*W(K)                                        MCSP0876
C                                                                       MCSP0877
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0878
      CCMMCN EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0881
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0882
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0883
```

```
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0885
      DIMENSION W(3)                                                     MCSP0886
C     ITRK NE.0 OR 1 -- SEVERAL TRACKS GENERATED,BUT NOT FROM STD.       MCSP0887
C                       LINEAR EQS.                                      MCSP0888
C          = 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS           MCSP0889
C          = 1 -- ONLY ONE TRACK IS USED                                 MCSP0890
      IF (ITRK.NE.0) GO TO 4                                             MCSP0891
      CALL SNORM (IW,W,IN)                                               MCSP0892
C     CONVERT EACH N(0,1) R.V. TO N(0,SIGW(I)) R.V.                      MCSP0893
C                                                                        MCSP0894
      DO 1 I=1,IN                                                        MCSP0895
    1 W(I) = SIGW(I)*W(I)                                                MCSP0896
C                                                                        MCSP0897
C                                                                        MCSP0898
      DO 2 I=1,N                                                         MCSP0899
    2 X(I) = XS(I,K)                                                     MCSP0900
C                                                                        MCSP0901
      CALL VPROD (GAMMAS,W,N,IN,W)                                       MCSP0902
      CALL VPROD (PHIS,X,N,N,VTMP)                                       MCSP0903
      CALL VADD (VTMP,W,N,VTMP)                                          MCSP0904
C                                                                        MCSP0905
      DO 3 I=1,N                                                         MCSP0906
    3 XS(I,K+1) = VTMP(I)                                                MCSP0907
C                                                                        MCSP0908
C     NEW VALUE OF X HAS BEEN COMPUTED AND STORED IN THE ARRAY XS        MCSP0909
      RETURN                                                             MCSP0910
    4 IF (ITRK.NE.1) GO TO 6                                             MCSP0911
C      IF(ITRK.EQ.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED   MCSP0912
C     TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY          MCSP0913
C     XS(I,K),I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE  MCSP0914
C     GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED       MCSP0915
C     IN XS(I,1),I=1,N)                                                  MCSP0916
C                                                                        MCSP0917
      TPI = 2.*3.14159265                                                MCSP0918
      DO 5 K=2,NSAM                                                      MCSP0919
      EKM1 = K-1
      T = 1.0*EKM1
      A=0.03333*T
      IF(A.LT.TPI) GO TO 10
      MM= A/TPI
      FM=MM
      A = A - FM*TPI
   10 CONTINUE
      XS(1,K)=10.*SIN(A)
      XS(2,K)=.3333*COS(A)
      XS(3,K)=10.*COS(A)
    5 XS(4,K)=-.3333*SIN(A)
```

```
C                                                                       MCSP0925
      RETURN                                                            MCSP0926
    6 CCNTINUE                                                          MCSP0927
C     IF THIS POINT IS REACHED, ITRK NCT EQUAL 0 CR 1 INDICATING THAT   MCSP0928
C     SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.     MCSP0929
C     LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE       MCSP0930
C     STATEMENTS HERE.                                                  MCSP0931
      RETURN                                                            MCSP0932
      END                                                               MCSP0933
      SUBROUTINE MEAS                                                   MCSP0734
C     THIS SUBROUTINE STARTS WITH THE TRUE STATE VALUE XS               MCSP0735
C     AND ADDS ZERO-MEAN WHITE GAUSSIAN NOISE TO H*XS TO                MCSP0736
C     GENERATE A NOISY VECTOR OF MEASUREMENTS Z.                        MCSP0737
C                                                                       MCSP0738
C                                                                       MCSP0739
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0740
      COMMCN EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0741
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0742
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0743
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0744
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0745
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0747
      ALPHA = XS(3,K)                                                   MCSP0748
      BETA = XS(1,K)                                                    MCSP0749
      Z(1) = SQRT(ALPHA**2+BETA**2)                                     MCSP0750
      Z(2) = ATAN2(ALPHA,BETA)                                          MCSP0751
      CALL SNCRM (IV,V,M)                                               MCSP0752
C                                                                       MCSP0753
      DO 1 I=1,M                                                        MCSP0754
    1 V(I) = SIGV(I)*V(I)                                               MCSP0755
C                                                                       MCSP0756
      CALL VADD (Z,V,M,Z)                                               MCSP0757
      ALPHA = Z(1)*COS(Z(2))                                            MCSP0758
      BETA = Z(1)*SIN(Z(2))                                             MCSP0759
      XZ(K)=ALPHA
      YZ(K)=BETA
      RETURN                                                            MCSP0762
      END                                                               MCSP0763
      SUBROUTINE GAIN                                                   MCSP0695
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0696
      CCMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0697
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0698
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0699
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0700
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0701
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0703
```

```
      DIMENSION BE(4),ER(4)
C                                                                 MCSP0704
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)                      MCSP0705
      DO 300 I=1,4
      DO 300 J=1,4
  300 PKKS(I,J,K)=PKKM1(I,J)
      PX(1)=0.
      PY(1)=0.
      PX(2)=10.
      PY(2)=10.
      IF(DABS(PKKM1(1,1)-PKKM1(3,3)).GT.0) GO TO 11
      PKKM1(1,1)=PKKM1(3,3)+0.000001
   11 CONTINUE
      THE=0.5*DATAN(2.*PKKM1(1,3)/(PKKM1(1,1)-PKKM1(3,3)))
      IF(ABS(THE).GT.0) GO TO 10
      THE = 0.00001
   10 CONTINUE
      SIG2X=(PKKM1(1,1)+PKKM1(3,3))/2.+PKKM1(1,3)/SIN(2.*THE)
      SIG2Y=(PKKM1(1,1)+PKKM1(3,3))/2.-PKKM1(1,3)/SIN(2.*THE)
      DO 9 IN=1,2
      IF(ABS(XHKKM1(1)-PX(IN)).GT.0) GO TO 9
      XHKKM1(1)=0.0000001+PX(IN)
    9 BE(IN)=ATAN((XHKKM1(3)-PY(IN))/(XHKKM1(1)-PX(IN)))
      IF(SIG2X.GE.SIG2Y) GO TO 63
      THE=THE+3.14159265/2.
   63 CONTINUE
   66 DO 4 IN=1,2
    4 ER(IN)= ABS(THE-BE(IN))
      IF(ABS(COS(ER(1))).LE.ABS(COS(ER(2)))) GO TO 7
      IN=1
      GO TO 8
    7 IN=2
    8 XIN=IN
        RR    =((XHKKM1(1)-PX(IN))**2+(XHKKM1(3)-PY(IN))**2)**.5
      WRITE (6,146) THE,SIG2X,SIG2Y,BE(1),BE(2),XIN,ER(1),ER(2), RR
  146 FORMAT (9(2X,1PE12.5),/)                                    MCSP0634
      H(1,1)=(XHKKM1(1)-PX(IN))/RR
      H(1,3)=(XHKKM1(3)-PY(IN))/RR
      CALL TRANS (H,M,N,TEMP2)                                    MCSP0706
      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)                          MCSP0707
      CALL PROD (H,TEMP,M,N,M,TEMP1)                              MCSP0708
      CALL ADD (TEMP1,R,M,M,TEMP1)                                MCSP0709
      IF (M.EQ.1) GO TO 2                                         MCSP0710
      MD = ND                                                     MCSP0711
      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)                      MCSP0712
      CALL PROD (TEMP,TEMP2,N,M,M,G)                              MCSP0713
C                                                                 MCSP0714
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE                         MCSP0715
```

```
C      P(K/K) = (I-G(K)*H)*P(K/K-1)                                    MCSP0716
     1 CALL PROD (G,H,N,M,N,TEMP)                                      MCSP0717
       CALL SUB (EI,TEMP,N,N,TEMP2)                                    MCSP0718
       CALL PROD (TEMP2,PKKM1,N,N,N,PKK)                               MCSP0719
C                                                                      MCSP0720
C      NOTE HERE  PKKM1(I,J) = P(K/K-1)  WHERE                         MCSP0721
C      P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q                               MCSP0722
       CALL TRANS (PHI,N,N,TEMP2)                                      MCSP0723
       CALL PROD (PKK,TEMP2,N,N,N,TEMP)                                MCSP0724
       CALL PROD (PHI,TEMP,N,N,N,TEMP1)                                MCSP0725
       CALL ADD (TEMP1,Q,N,N,PKKM1)                                    MCSP0726
       RETURN                                                          MCSP0727
C                                                                      MCSP0728
     2 DO 3 I=1,N                                                      MCSP0729
     3 G(I,1) = TEMP(I,1)/TEMP1(1,1)                                   MCSP0730
C                                                                      MCSP0731
       GO TO 1                                                         MCSP0732
       END                                                             MCSP0733
       SUBROUTINE ESTIM                                                MCSP0655
C      THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT      MCSP0656
C      CONDITION (IEST.EQ.0) THE STANDARD EQUATIONS                    MCSP0657
C                                                                      MCSP0658
C      XHAT(K/K)=XHAT(K/K-1)+G(K)*(Z(K)-H(K)*XHAT(K/K-1))              MCSP0659
C                                                                      MCSP0660
C      XHAT(K+1/K)=PHI*XHAT(K/K)                                       MCSP0661
C                                                                      MCSP0662
C      ARE EVALUATED                                                   MCSP0663
       REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI     MCSP0664
       COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),     MCSP0665
      1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
      2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),        MCSP0667
      3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    MCSP0668
      4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4), MCSP0669
      5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
      6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND         MCSP0671
C      TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK             MCSP0672
C                                                                      MCSP0673
C                                                                      MCSP0675
       XIN=IN
       DO 1 I=1,N                                                      MCSP0674
       DO 1 J=1,M                                                      MCSP0676
     1 GK(I,J) = GKS(I,J,K)                                            MCSP0677
C                                                                      MCSP0678
       IF (IEST.NE.0) GO TO 2                                          MCSP0679
       CALL VPROD (HS,XHKKM1,M,N,VTMP)                                 MCSP0680
       VTMP(1)=((XHKKM1(1)-PX(IN))**2+(XHKKM1(3)-PY(IN))**2)**.5
       Z(1)= ((XZ(K)-PX(IN))**2+(YZ(K)-PY(IN))**2)**.5
       CALL VSUB (Z,VTMP,M,VTMP)                                       MCSP0681
```

```
      CALL VPROD (GK,VTMP,N,M,VTMP)                                     MCSP0682
      CALL VADD (XHKKM1,VTMP,N,XHKK)                                    MCSP0683
C                                                                       MCSP0684
C     XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK          MCSP0685
      CALL VPROD (PHIS,XHKK,N,N,XHKKM1)                                 MCSP0686
C                                                                       MCSP0687
C     XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1      MCSP0688
      RETURN                                                            MCSP0689
    2 CONTINUE                                                          MCSP0690
C     IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE         MCSP0691
C     EQUATIONS MUST BE INSERTED HERE BY THE USER.                      MCSP0692
      RETURN                                                            MCSP0693
      END                                                               MCSP0694
      SUBROUTINE PRT
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI       MCSP0878
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),          MCSP0881
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      MCSP0882
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0883
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND           MCSP0885
      IG=0
      ISTAT=1
      WRITE (6,147)                                                     MCSP0446
C     WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR          MCSP0447
C     IF ONE SET OF GAINS HAS BEEN USED.                                MCSP0448
      IF (IG.NE.0) GO TO 61                                             MCSP0449
      WRITE (6,148)                                                     MCSP0450
C                                                                       MCSP0451
      DO 59 K=1,NSAM                                                    MCSP0452
      WRITE (6,149) K                                                   MCSP0453
C                                                                       MCSP0454
      DO 59 I=1,N                                                       MCSP0455
   59 WRITE (6,146) (GKS(I,J,K),J=1,M)                                  MCSP0456
C                                                                       MCSP0457
      WRITE (6,150)                                                     MCSP0458
C                                                                       MCSP0459
      DO 60 K=1,NSAM                                                    MCSP0460
      WRITE (6,151) K                                                   MCSP0461
C                                                                       MCSP0462
      DO 60 I=1,N                                                       MCSP0463
   60 WRITE (6,146) (PKKS(I,J,K),J=1,N)                                 MCSP0464
C                                                                       MCSP0465
C                                                                       MCSP0466
C                                                                       MCSP0467
   61 WRITE (6,156)                                                     MCSP0468
      WRITE (6,152)                                                     MCSP0469
```

```
      WRITE (6,153)                                                    MCSP0470
C                                                                      MCSP0471
      DO 62 K=1,NSAM                                                   MCSP0472
      WRITE (6,155)                                                    MCSP0473
C                                                                      MCSP0474
      DO 62 I=1,N                                                      MCSP0475
   62 WRITE (6,154) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)                    MCSP0476
C                                                                      MCSP0477
      WRITE (6,156)                                                    MCSP0478
      IF (ISTAT.EQ.0) GO TO 64                                         MCSP0479
      WRITE (6,157)                                                    MCSP0480
C                                                                      MCSP0481
      DO 63 K=1,NSAM                                                   MCSP0482
      WRITE (6,158) K                                                  MCSP0483
C                                                                      MCSP0484
      DO 63 I=1,N                                                      MCSP0485
   63 WRITE (6,146) (VAR(I,L,K),L=1,I)                                 MCSP0486
   64 CONTINUE
C                                                                      MCSP0487
      WRITE (6,156)                                                    MCSP0488
  146 FORMAT (9(2X,1PE12.5),/)
  147 FORMAT ('1',20X,'OUTPUT DATA',///)                               MCSP0635
  148 FORMAT (10X,'THE GAIN MATRICES ARE',/)                           MCSP0636
  149 FORMAT (5X,'K=',I3,/,10X,'G(K)=',/)                              MCSP0637
  150 FORMAT (1X,//,10X,'THE THEORETICAL COVARIANCE MATRIX IS',/)      MCSP0638
  151 FORMAT (5X,'K=',I3,/,10X,'P(K/K)=',/)                            MCSP0639
  152 FORMAT (T5,'TIME',T16,'VECTOR COM-',T34,'SAMPLE MEAN',           MCSP0640
     1 T51,'SAMPLE MEAN OF',T71,'SAMPLE VARIANCE OF')                  MCSP0641
  153 FORMAT (T5,'INDEX',T16,'PONENT INDEX',T34,'OF TRACK',            MCSP0642
     1 T51,'ESTIMATION ERROR',T71,'ESTIMATION ERROR')                  MCSP0643
  154 FORMAT (6X,I3,13X,I1,10X,1PE14.7,2(6X,1PE14.7))                  MCSP0644
  155 FORMAT (//)                                                      MCSP0645
  156 FORMAT ('1')                                                     MCSP0646
  157 FORMAT (10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS',//)  MCSP0647
  158 FORMAT (//,2X,'K=',I3,/)                                         MCSP0648
      RETURN
      END
      SUBROUTINE PLT
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI      MCSP0878
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),      MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0880
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,60),ERR(4,60),         MCSP0881
     3GAMMAS(4,4),PHIS(4,4),XS(4,60),HS(4,4),GK(4,4),SIGW(4),X(4),     MCSP0882
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),  MCSP0883
     5XHATZ(4),XZ(60),YZ(60),PX(10),PY(10),
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND          MCSP0885
      DIMENSION XP(60),YP(60)
      INTEGER*4 ITB(12)/12*0/
```

```
      REAL*4     RTB(28)/28*0.0/
      EQUIVALENCE (TITLE,RTB(5))
      REAL*8 TITLE(12)/'X = TRUE,   + = FILTER,   SQUARE = NOISY'/
      IGPLT=1
      ITHVPL=1
      IMTPLT=1
      ISMPLT=1
      ISVPLT=1
      DO 50 K=1,NSAM
      XP(K)=XM(1,K)
   50 YP(K)=XM(3,K)
      CALL PLOTP(XP,YP,NSAM,0)
      ITB(1)=1
      ITB(2)=1
      CALL DRAWP(60,XP,YP,ITB,RTB)
      DO 51 K=1,NSAM
      XP(K)=XS(1,K)+ERR(1,K)
   51 YP(K)=XS(3,K)+ERR(3,K)
      CALL PLOTP(XP,YP,NSAM,0)
      ITB(1)=2
      ITB(2)=2
      CALL DRAWP(60,XP,YP,ITB,RTB)
      ITB(2)=0
      DO 2 J=1,60,5
      IF(ABS(PKKS(1,1,J)-PKKS(3,3,J)).GT.0) GO TO 11
      PKKS(1,1,J)=PKKS(3,3,J)+0.000001
   11 CONTINUE
      THE=0.5*ATAN(2.*PKKS(1,3,J)/(PKKS(1,1,J)-PKKS(3,3,J)))
      IF(ABS(THE).GT.0) GO TO 10
      THE=0.000001
   10 CONTINUE
      SIG2X=(PKKS(1,1,J)+PKKS(3,3,J))/2.+PKKS(1,3,J)/SIN(2.*THE)
      SIG2Y=(PKKS(1,1,J)+PKKS(3,3,J))/2.-PKKS(1,3,J)/SIN(2.*THE)
      WRITE (6,146) THE,SIG2X,SIG2Y
  146 FORMAT (9(2X,1PE12.5),/)
      SX=(SIG2X)**.5*20.
      SY=(SIG2Y)**.5*20.
      PT=3.14159265/12.
      CT=COS( THE )
      ST=SIN( THE )
      DO 1 I=1,25
      XI=I
      XP(I)=SX*COS( PT*XI)*CT-SY*SIN(PT*XI)*ST+XS(1,J)
    1 YP(I)=SX*COS( PT*XI)*ST+SY*SIN(PT*XI)*CT+ XS(3,J)
    2 CALL DRAWP(25,XP,YP,ITB,RTB)
      ITB(1)=3
      ITB(2)=3
      CALL DRAWP(60,XZ,YZ,ITB,RTB)
```

```
      DO 65 K=1,NSAM                                                    MCSP0491
   65 XP(K) = K                                                         MCSP0492
C                                                                       MCSP0493
      IF (IGPLT.NE.1) GO TO 68                                          MCSP0494
C                                                                       MCSP0495
      DC 67 I=1,N                                                       MCSP0496
C                                                                       MCSP0497
      DO 67 J=1,M                                                       MCSP0498
C                                                                       MCSP0499
      DO 66 K=1,NSAM                                                    MCSP0500
   66 YP(K) = GKS(I,J,K)                                                MCSP0501
C                                                                       MCSP0502
      WRITE (6,156)                                                     MCSP0503
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0504
   67 WRITE (6,159) I,J                                                 MCSP0505
C                                                                       MCSP0506
   68 IF (ITHVPL.NE.1) GO TO 71                                         MCSP0507
C                                                                       MCSP0508
      DO 70 I=1,N                                                       MCSP0509
C                                                                       MCSP0510
      DO 69 K=1,NSAM                                                    MCSP0511
   69 YP(K) = PKKS(I,I,K)                                               MCSP0512
C                                                                       MCSP0513
      WRITE (6,156)                                                     MCSP0514
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0515
   70 WRITE (6,160) I,I                                                 MCSP0516
C                                                                       MCSP0517
   71 IF (IMTPLT.NE.1) GO TO 74                                         MCSP0518
C                                                                       MCSP0519
      DO 73 I=1,N                                                       MCSP0520
C                                                                       MCSP0521
      DO 72 K=1,NSAM                                                    MCSP0522
   72 YP(K) = XM(I,K)                                                   MCSP0523
C                                                                       MCSP0524
      WRITE (6,156)                                                     MCSP0525
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0526
   73 WRITE (6,161) I                                                   MCSP0527
C                                                                       MCSP0528
   74 IF (ISMPLT.NE.1) GO TO 77                                         MCSP0529
C                                                                       MCSP0530
      DC 76 I=1,N                                                       MCSP0531
C                                                                       MCSP0532
      DO 75 K=1,NSAM                                                    MCSP0533
   75 YP(K) = ERR(I,K)                                                  MCSP0534
C                                                                       MCSP0535
      WRITE (6,156)                                                     MCSP0536
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0537
   76 WRITE (6,162) I,I                                                 MCSP0538
```

```
C                                                                  MCSP0539
   77 IF (ISVPLT.NE.1) GO TO 80                                    MCSP0540
C                                                                  MCSP0541
      DO 79 I=1,N                                                  MCSP0542
C                                                                  MCSP0543
      DO 78 K=1,NSAM                                               MCSP0544
   78 YP(K) = VAR(I,I,K)                                           MCSP0545
C                                                                  MCSP0546
      WRITE (6,156)                                                MCSP0547
      CALL PLOTP (XP,YP,NSAM,0)                                    MCSP0548
   79 WRITE (6,163) I                                              MCSP0549
   80 CONTINUE
C                                                                  MCSP0550
      WRITE (6,156)                                                MCSP0551
  156 FORMAT ('1')
  159 FORMAT (12X,'G(',I1,',',I1,')  VS. K')                       MCSP0649
  160 FORMAT (12X,'PKK(',I1,',',I1,')  VS. K')                     MCSP0650
  161 FORMAT (12X,'MEAN OF X(',I1,')  VS. K')                      MCSP0651
  162 FORMAT (12X,'XHATKK(',I1,') -X(',I1,')  VS. K')              MCSP0652
  163 FORMAT (12X,'ERROR VARIANCE(',I1,') VS.. K')                 MCSP0653
      RETURN
      END
//LINK.USDD DD DSNAME=F0131.TITUS,UNIT=2314,DISP=SHR,
// VOLUME=SER=DUFFY
//LINK.SYSIN DD *
 INCLUDE USDD(PLRS)
 ENTRY MAIN
//GO.FTC6F001 DD SPACE=(CYL,(5,1))
//GO.SYSIN DD *
         4         1         2         60         1
 4
         1                             1                   1         1

                                    1         1
1.0         1.0
            1.0
                      1.0         1.0
                                  1.0
1.0
0.0001
.0001
             .0001
0.5
1.0
            0.5
            1.0
.0001                  .0001
            .0001
```

.0001		.0001		
			.0001	
.01	0.3333	10.		
		0.3333		10.

LIST OF REFERENCES

1. H. W. Sorenson, "Kalman Filtering Techniques," <u>Advan. Control System</u>, Vol. 3, Chapter 5, 1966.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H.A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	18
5.	Capt. Charles Dittmar Weapons and Systems Engineering United States Naval Academy Annapolis, Maryland	1
6.	Assoc. Professor Donald E. Kirk, Code 52Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Commander Naval Electronics System Command National City Washington D.C. 22030 ATTN: Charles Gill	2
8.	Marine Corps Test and Evaluation Center Camp Pendleton, California ATTN: Major Norbert Spitzer	2
9.	Robert R. Fossum Dean of Research (Code 023) Naval Postgraduate School Monterey, California 93940	1